Report Number: WUCSE-2002-44

2002-12-12

Reconfiguration in an Optical Multiring Interconnection Network - Masters Thesis, December 2002

Praveen Krishnamurthy

The advent of optical technology that can feasibly support extremely high bandwidth chip-to-chip communication raises a host of architectural questions in the design of digital systems. Terabit per second (and higher) bandwidths have not been previously available at the chip level. In this thesis, we examine the use of this technology in two different scenarios, viz., as the interconnection network in a multiprocessor system and as a switch fabric in network routers. Specifically, we examine the performance gains associated with utilizing the bandwidth reconfiguration capabilities of a system based on this technology.
... Read complete abstract on page 2.

Reconfiguration in an Optical Multiring Interconnection Network - Masters Thesis, December 2002

Praveen Krishnamurthy

Complete Abstract:

The advent of optical technology that can feasibly support extremely high bandwidth chip-to-chip communication raises a host of architectural questions in the design of digital systems. Terabit per second (and higher) bandwidths have not been previously available at the chip level. In this thesis, we examine the use of this technology in two different scenarios, viz., as the interconnection network in a multiprocessor system and as a switch fabric in network routers. Specifically, we examine the performance gains associated with utilizing the bandwidth reconfiguration capabilities of a system based on this technology.

Department of Computer Science & Engineering

Washington University in St. Louis
School of Engineering & Applied Science

2002-44

Reconfiguration in an Optical Multiring Interconnection Network - Masters Thesis, December 2002

Authors: Krishnamurthy, Praveen

December 12, 2002

Abstract: The advent of optical technology that can feasibly support extremely high bandwidth chip-to-chip communication raises a host of architectural questions in the design of digital systems. Terabit per second (and higher) bandwidths have not been previously available at the chip level. In this thesis, we examine the use of this technology in two di erent scenarios, viz., as the interconnection network in a multiprocessor system and as a switch fabric in network routers. Speci cally, we examine the performance gains associated with utilizing the bandwidth recon guration capabilities of a system based on this technology.

Reconfiguration in an Optical Multiring Interconnection Network

Praveen Krishnamurthy

Praveen Krishnamurthy, “Reconfiguration in an Optical Multiring Interconnection Network,” Master’s Thesis, Technical Report WUCS-2002-44, Department of Computer Science and Engineering, Washington University, Saint Louis, MO, 2002.

Computer and Communications Research Center
Washington University
Campus Box 1115
One Brookings Dr.
St. Louis, MO 63130-4899

WASHINGTON UNIVERSITY

SEVER INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

RECONFIGURATION IN AN OPTICAL MULTIRING INTERCONNECTION NETWORK

by

Praveen Krishnamurthy

Prepared under the direction of Professor Roger D. Chamberlain

---

A thesis presented to the Sever Institute of
Washington University in partial fulfillment
of the requirements for the degree of

Master of Science

December, 2002

Saint Louis, Missouri

WASHINGTON UNIVERSITY
SEVER INSTITUTE OF TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ABSTRACT

RECONFIGURATION IN AN OPTICAL MULTIRING INTERCONNECTION NETWORK

by Praveen Krishnamurthy

ADVISOR: Professor Roger D. Chamberlain

December, 2002
Saint Louis, Missouri

The advent of optical technology that can feasibly support extremely high bandwidth chip-to-chip communication raises a host of architectural questions in the design of digital systems. Terabit per second (and higher) bandwidths have not been previously available at the chip level. In this thesis, we examine the use of this technology in two different scenarios, viz., as the interconnection network in a multiprocessor system and as a switch fabric in network routers. Specifically, we examine the performance gains associated with utilizing the bandwidth reconfiguration capabilities of a system based on this technology.

Contents

List of Tables . . . . . **iv**

List of Figures . . . . . **v**

Acknowledgments . . . . . **vii**

1 Introduction **1**

1.1 Overview 1

1.2 Background and Related Work 2

1.2.1 Optics in parallel Computer systems 2

1.2.2 Optics in Networking 3

1.2.3 Reconfigurability in Optical Systems 4

1.2.4 Communication Requirements in Parallel Programs 5

1.2.5 Traffic Models in Networking Systems 5

1.3 Goals and Contributions 5

1.4 Organization of thesis 7

2 The Hardware System and Simulation Model **8**

2.1 Physical Devices 8

2.2 System Descriptions 11

2.2.1 Network Architecture of the Multiring 11

2.2.2 Media Access Protocol 12

2.2.3 DRR Scheduling 13

2.3 Performance Characteristics of the Multiring 17

2.4 rICNS : A MODSIM III based Simulator for the Multiring 17

2.4.1 Simulation Model for the Multiring 18

3 System Reconfiguration **20**
3.1 Introduction 20
3.2 Motivation for Reconfiguration 21
3.2.1 Parallel Computation 21
3.2.2 Internet switching 21
3.3 Physical Techniques for Reconfiguration 22
3.3.1 Reconfiguration among the subrings 22
3.3.2 Reconfiguration within a subring 24
3.3.3 Summary of Physical Mechanisms 25
3.4 Static Reconfiguration - Description 25
3.4.1 Simulation Model Details 26
3.5 Dynamic Reconfiguration - Description 33
3.5.1 Simulation Model Details 34

4 Simulation Experiments and Results **39**
4.1 Introduction 39
4.2 Static Reconfiguration 40
4.2.1 Real Applications 40
4.2.2 Synthetic Applications 40
4.2.3 Performance Analysis 40
4.3 Dynamic Reconfiguration Results 45
4.3.1 Performance Analysis 47
4.3.2 Reconfiguration at sub-optimal period 57
4.4 Summary 61

5 Conclusion and Future work **62**
5.1 Contributions 62
5.1.1 Static Reconfiguration 62
5.1.2 Dynamic Reconfiguration 63
5.2 Summary 63
5.3 Future Work 64

Vita **70**

List of Tables

2.1 Brief description of key objects in the simulator 19

3.1 Details of the synthetic applications simulated 29
3.2 Details of the synthetic applications simulated contd. 30
3.3 Details of the synthetic applications simulated contd. 31

4.1 Completion time data for the 12 applications simulated 45
4.2 Mean queue lengths (cells) 48
4.3 Overall standard deviation over all runs 53
4.4 Reconfiguration penalty per packet (R_c) (comparison between simulation model and Little's Law) 59

List of Figures

2.1 A Smart Pixel 9
2.2 Optical I/O at chip level 9
2.3a Rigid free-space optical link 10
2.3b Fiber image guide optical link 10
2.4 A four chip optical ring topology 11
2.5 Multiring Topology - Showing logically separate channels 12
2.6 Allocation of VCSEL-detector pairs to a four channel system 13
2.7 Conceptual diagram of a 4 node multiring 14
2.8 Channel 4 of a 4 node ring 15
2.9 Placement of DRR fairness layer 15
2.10 Model of the ring used in simulation 18

3.1a Uniform allocation of VCSELs to channels 23
3.1b Reconfigured allocation of VCSELs to channels 24
3.2 LCA reconfiguration at each node 25
3.3 SAR phases. 26
3.4a Broadcast from sensor node to compute nodes. 27
3.4b Corner turn between compute nodes. 28
3.4c Reduction from compute nodes to output node. 28
3.5 Multiring technology as part of a router switch 35
3.6a Batch mean for 100% utilization - subring 3. 37
3.6b Batch mean for 100% utilization - Across all sources for a given destination. 37
3.7a Batch mean for 50% utilization - subring 3. 38
3.7b Batch mean for 50% utilization - Across all sources for a given destination. 38

4.1 Maximum and mean completion time for point-to-point communication phases . . . 42
4.2 Variability in completion times for point-to-point communication phases 43
4.3 Communication phase completion times across applications (with and without reconfiguration) . . . 44
4.4 Maximum, median and minimum speedup obtained across communication patterns. . . . 46
4.5 Overall performance improvement. . . . 47
4.6a Uniform Allocation . . . 49
4.6b Reconfigured - Million celltime period . . . 49
4.6c Reconfigured - Hundred thousand celltime period . . . 50
4.6d Reconfigured - Thousand celltime period . . . 50
4.7a Average delay in packet delivery in the system (overall) . . . 52
4.7b Average delay in packet delivery in the system (subring) . . . 52
4.8a Standard deviation of packet delay across all subrings . . . 54
4.8b Standard deviation of packet delay in individual subrings . . . 54
4.9a Average Packet Delay for each source in each subring (Uniform) . . . 55
4.9b Average Packet Delay for each source in each subring (DRR only) . . . 55
4.10aAverage Message Delay on each subring (Uniform) . . . 56
4.10bAverage Message Delay on each subring (LCA only) . . . 56
4.11 Average packet delay across all subrings . . . 60

Acknowledgments

Many people have helped me during the course of this thesis. Dr. Roger Chamberlain has provided a motivating, and enthusiastic atmosphere during the course of this work. I would like to thank him for being a great advisor. I would like to thank Dr. Mark Franklin who kept an eye on the progress of my work and was always available when I needed his help. I would also like to thank Dr. Jason Fritts for being on my thesis committee, and also providing me with valuable comments on earlier versions of this thesis. Thanks to Dr. Daniel Fuhrmann, for answering questions related to this thesis.

This research was supported in part by DARPA under grant DAAL01-98-C-0074 and the NSF under grants CCR-0217334 and ACI-0203869. I am grateful to these institutions for their support.

I would like to thank David Zar and Allen Rueter for their help with computers. I would also like to thank Paula Hardy-Mumm, Peggy Fuller, Jean Grothe, Myrna Harbison, Sharon Matlock, and Ouida Jackson for helping me with the administrative side of things.

My colleagues here at Washington University made it easier for me to adjust to this work environment, and have been fun to work with. I would like to thank all of them for their support.

I will always be grateful to my parents for their love, and belief in my abilities. I thank them and the rest of my family for their encouragement.

Praveen Krishnamurthy

Washington University in Saint Louis
December 2002

Chapter 1

Introduction

1.1 Overview

A significant aspect of parallel computation is data communication between the various processors in the system. Parallel computer systems with a large number of processors can significantly improve the performance of many applications. Advances in silicon-based technologies have increased processor speeds into the gigahertz domain and decreased the per processor cost considerably. These factors have contributed to a significant increase in the use of parallel machines. This places a very high demand on the interconnection network, to the point where interconnection technology is the performance bottleneck in many parallel systems.

The high bandwidth of optics makes it ideally suited to form the interconnection network in these system, provided the implementation complexities can be managed. A system based on optics as its interconnection network has been introduced in [11, 28] and is briefly described in the next chapter. This thesis presents the benefits of *reconfigurability* in such an optically interconnected system. The *multiring* architecture [11] is extended to be a reconfigurable architecture with the ability to change bandwidth allocation at runtime. Parts of this thesis have already been published [10].

Two distinct types of reconfiguring the multiring interconnect viz., *Static Reconfiguration* and *Dynamic Reconfiguration* are presented, pertaining to different classes of applications. The former corresponding to reconfiguration in that is appropriate for signal processing applications with *a priori* knowledge of the communication requirements, where as the later method is applied to a network switch fabric where prior knowledge of requirements is unknown.

A simulator based on the ICNS framework [8] has been modified to support *reconfigurability*. The performance implications of using *static reconfiguration* and *dynamic reconfiguration* in an 8 node multicomputer system and an 8 port network switch are presented.

1.2 Background and Related Work

This section presents an overview of optical communication in general. It describes the use of optics in a multi-computer environment and also describes its use in a large scale network such as the network.

1.2.1 Optics in parallel Computer systems

With advances in VLSI technology, processing speed has grown much faster than the communication bandwidth supported by the interconnection infrastructure in multicomputer systems, thus creating a mismatch bottleneck in the interconnection network [31]. Research has shown that the performance of Massively Parallel Processing systems (MPPs) is significantly dependent on the underlying interconnection network.

The idea of using optics as an interconnection network in parallel multicomputers has been around for some years now. The inherent advantages in using optics such as reduced crosstalk, low power requirements, better isolation compared to semiconductor or metal interconnection, and primarily high speed have been the main motivation behind the interest in this technology. The main deterrent against its use has been the cost of implementing such system. However, recent work [18] shows that vertical cavity surface emitting laser (VCSEL) [24] based interconnections are becoming cost competitive with metal interconnections.

A significant cost benefit of VCSELs is their ability to form arrays. They also differ from edge-emitting lasers substantially. Conventional edge-emitters, which release light from their side (parallel to the substrate), have numerous drawbacks in cost, manufacturability, and reliability. These drawbacks are the result of the manufacturing process, which does not allow for the lasers to be tested until they have been cleaved and packaged. VCSELs, on the other hand, can be tested for their reliability and functionality on the wafer.

There have been a number of previous designs proposed to exploit optics in multicomputer interconnects. The *Gemini* project [9], for example, is a tightly-coupled multi-computer system which exploits optics in the interconnection network. This system contains an optical data path (with switching performed in the optical domain via $LiNbO_3$ switching elements) and an electrical control path. This dual architecture ensures a high bandwidth in the data path and exploits the benefits of using electronics for logic and control.

Work by Rami Melhem's team [44, 43] describes the use of optical technology in large scale parallel processing systems. They propose the use of time division multiplexing (TDM) for improving the performance of these optical interconnection networks in general. The use of VCSELs in a massively parallel processor system has been described in [20]. They suggest implementing an *ultra* dense optical interconnection network for massively parallel processors, using two dimensional arrays of beam steering VCSELs. They also suggest using space division switching elements in a free space photonic Banyan (or other multistage) network.

An advantage of using VCSELs is that no external power source is need to power the optical emissions. Also, the VCSELs are compact compared to edge emitting lasers and have a low threshold current, which in turn decreases the power consumption. A prototype board-to-board interconnection network based on VCSEL technology has also been built [39].

One VCSEL property is its ability to be laid out as a two dimensional array. Applications of this technology (i.e., VCSELs and free-space communication in interconnection networks) have been discussed in [48]. The commercial feasibility of VCSELs in opto-electronic interconnect technologies is evident from the $Teralink^{\text{TM}}24$ and $Teralink^{\text{TM}}48$ series of interconnect modules from Teraconnect Inc. [23]. Their product uses a two dimensional array of VCSELs with 24 or 48 channels and has an aggregate data bandwidth of 76 and 150 Gb/s respectively.

With increases in parallelism and the development of smart pixel technology [22], the role of free-space optical communication between logic elements becomes a more feasible option compared to bulk optics or even fiber transmission.

1.2.2 Optics in Networking

One of the major issues the networking industry faces is meeting the continually increasing bandwidth requirements. Optical networks address this issue with high

bandwidth link solutions. Significant additional benefits can be attributed to the development of Wavelength Division Multiplexing(WDM) and Dense Wavelength Division Multiplexing (DWDM) [21], which provide additional capacity in the existing fiber optics channels.

The above advances focus on link technologies. Here our interest is on switching technology. Switches are responsible for connecting links at a low-level network protocol layer. Technically switches operate at layer two of the OSI model.

Traditionally switching has been in done in the electrical/electronic domain. A packet/frame/cell is received on a link, the header information is extracted, a routing or forwarding decision is made to determine the outgoing link based on the header information, and the packet/frame/cell is delivered to the chosen outgoing link. Standard Ethernet switches use a media access control (MAC) address on the frame and makes in the forwarding decision based on this information. Likewise multiprotocol label switching (MLPS) Label Switch Routers use the outermost label to make their forwarding decision. As most of the optical switches will be used in DWDM installations, attempts are being made to make forwarding decisions based on the wavelength (per-wavelength switches).

There has been a great amount of research focused on *all-optical switching* which eliminates the optical to electronic signal conversion and vise-versa. Research has also focused on identifying suitable architectures (like ring, mesh, multiring [32], etc.) and also on routing issues [1, 36]. Our approach here is a dual approach, where we use the optical domain to transmit the data chip-to-chip, but switching will still be performed in the electronic domain. This thesis explores reconfiguration in such a system.

1.2.3 Reconfigurability in Optical Systems

Advances in optical technology have not only paved the way for optical interconnections at different levels viz., chip-to-chip, board-to-board, node-to-node, etc., but also pose challenges for maximizing the available resources. Merely substituting the metal interconnections with optics does not make use of the high parallelism and bandwidth efficiently.

Reconfigurability gives us the ability to use these resources effectively. Qiao and Melhem [41] describe the benefits of reconfiguring an optical interconnection network using Time Division Multiplexing when the requirements of the application

running on the multiprocessors are known ahead of time. In this case they go about the process of repeatedly changing the mapping of the multi-stage interconnection networking in a time division methodology. They also talk of ways of dynamically reconfiguring an electro-optical switch in a multiprocessor environment using Time Division Multiplexing [43]. As part of this thesis we consider a multiring architecture, which is reconfigured to the needs of the application. Per-flow bandwidth of this fully connected system is reconfigured by using two reconfiguration techniques viz., Laser Channel Allocation (LCA) and Deficit Round Robin Allocation (DRR). We evaluate the benefits of such a reconfigurable system in a multicomputer environment and also in a broader networking system.

1.2.4 Communication Requirements in Parallel Programs

Communication in parallel systems often follows common patterns. These patterns can be classified into four major types viz., All-to-All, Broadcast, Reduce and Point-to-Point. Also with parallel programs such as those in signal processing applications, the communication requirements are known *a priori*. These applications tend to be characterized by alternating communication and computation phases, which gives us the opportunity to reconfigure the interconnection network during the computation phases [5].

1.2.5 Traffic Models in Networking Systems

There have a been numerous models suggested to represent the traffic patterns on the internet. Earlier models characterized the interarrival time between packets to be an exponential distribution, i.e., a Poisson arrival process. This model, though is it valid for modeling user sessions such as terminals, fails to be accurate for Wide Area Networks (WAN) [38]. Later work has shown that the internet traffic is *self-similar* [17]. Self-similarity is a property which implies that the object (in this case the distribution of the traffic) looks the same, even with varying time scales.

1.3 Goals and Contributions

As part of the work in this thesis we want to model a reconfigurable interconnection network which is flexible enough to allocate bandwidth on a per flow basis. The initial focus is to develop a model which serves as a platform for signal processing

applications (where the bandwidth requirements are known *a priori*), the concept is then extended to a network switching fabric with unpredictable load.

In the multicomputer environment we want to establish the overall benefits of reconfiguring the interconnection network to suit the communication requirements of applications. The knowledge of the communication requirements in this case makes it easy for us to establish a communication configuration for the interconnection network optimized for the requirements of each communication phase. Choosing a configuration becomes an interesting problem in the switch fabric case, where such requirements are not known. We investigate different periods of time between reconfiguring such a system, with the objective of decreasing the overall delay and maintaining a desired degree of fairness between all the ports of the switch.

The following list enumerates the specific contribution of the thesis:

- Enhanced the ICNS framework to simulate a true multiring and to support reconfiguration.
- **Static Reconfiguration**
  - Identified the characteristics of applications that can benefit from a statically reconfigured interconnection network.
  - Developed models of both real and synthetic application's communication requirements.
  - Modeled the communications of the applications set (both real and synthetic) via simulation.
  - Used analytical models to put communications performance in the context of overall application performance.
- **Dynamic Reconfiguration**
  - Developed an input model to generate *self-similar* input traffic for a simulated switching system.
  - Implemented a dynamic control algorithm for reconfiguring the switch based on input backlog.
  - Obtained performance numbers for various traffic loads via simulation.
  - Used analytical models to include reconfiguration cost in the performance results.

1.4 Organization of thesis

With the main objectives of this work defined, this section gives the organization of the content of the thesis. Chapter 2 describes the hardware system where issues like architecture, media access protocol and the DRR fairness protocol are discussed. The chapter also goes on to describe the ICNS-based multiring simulator.

Chapter 3 delves into the idea of reconfiguration in this system. It describes the types of reconfiguration viz., *Static Reconfiguration* and *Dynamic Reconfiguration*, and the applications where they are applicable. It also describes methods of reconfiguration such as *Laser Channel Allocation* and *Deficit Round Robin Allocation* as well as the control algorithm employed for dynamic reconfiguration. This chapter also describes the features of the simulation model used for exploring the benefits of reconfiguration in the various classes of applications considered.

After this we proceed to describe the performance results in Chapter 4. Benefits of each method of reconfiguration are discussed, and also their combined effect on applications for the static reconfiguration is presented. An analytical model using Amdahl's law is presented for obtained overall performance numbers including the computation phases for the applications in static reconfiguration. For the dynamic reconfiguration case, the chapter discusses the reasoning for choosing a period for reconfiguring the switch. Performance numbers are presented for the various reconfiguration periods and are compared to the uniform allocation case. Conclusions from the results obtained in this work are summarized in Chapter 5.

Chapter 2

The Hardware System and Simulation Model

This chapter provides the description for the various physical devices that form the basic components in this research. It also describes the system architecture that is the subject of the performance analysis. It describes components such as the Vertical Cavity Surface Emitting Lasers (VCSELs) and the Metal Semiconductor Metal (MSM) photodetectors. These are the core elements that form the high bandwidth optical interconnect, capable of providing terabits per second of bandwidth for interchip (i.e., between two processors or between processor and memory) communication.

The chapter also describes certain system characteristics such as network topology channel design and Deficit Round Robin (DRR) scheduling. It also gives some performance characteristics of the system with and without the presence of the DRR scheduling. The later part of this chapter will describe the rICNS, the Multiring Interconnect Network Simulator Program, used to model the optical interconnect. Some of the material in this chapter is derived from [28].

2.1 Physical Devices

At the core of the interconnect reside arrays of "Smart Pixels." A smart pixel is an optoelectronic structure composed of an electronic processing circuit (CMOS, BiCMOS, bipolar, etc.) enhanced with optical inputs and/or outputs (Figure 2.1). The optical outputs use VCSELs for electrical-to-optical signal conversion and the optical inputs

are sensed by either MSM detectors or photodiodes providing optical-to-electrical signal conversion. A two dimensional arrangement of these elements is referred to as a Smart Pixel Array (SPA).

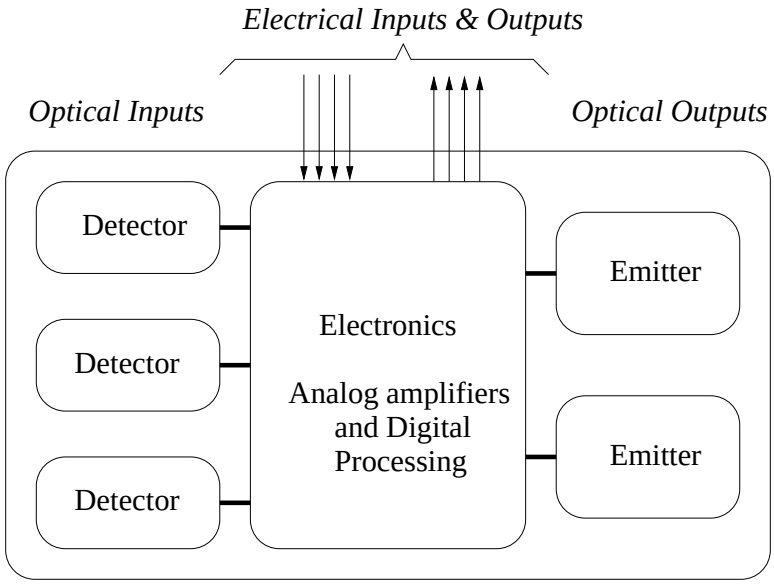

Figure 2.1: A Smart Pixel

Figure 2.2 illustrates the arrangement of a 2 x 2 SPA, i.e., a chip having a 2 x 2 array of VCSELs and a 2 x 2 array of detectors. We can also see in Figure 2.2 that the VCSELs transmit light perpendicular to the plane of the chip. Chips with

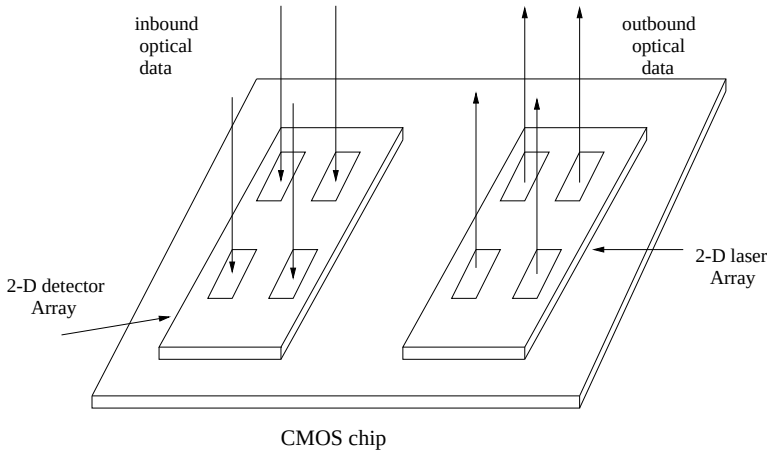

Figure 2.2: Optical I/O at chip level

an integrated SPA can be interconnected with the help of optical components such as mirrors and lenses (Figure 2.3a), or a more versatile fiber optic image guide (Figure 2.3b) to form a network of optically interconnected chips. Designs incorporating

the former can be found in [4, 40] and more description on the latter can be found in [19, 30]. While the demonstration of [39] used bulk optics to deliver light between ICs, designs have been investigated utilizing both rigid optical links [12] optimized to be misalignment tolerant (useful for chip-to-chip links on a board), and flexible fiber imaging guides [27] (useful for board-to-board links). Given the vertical nature of the VCSEL process, both approaches require connection to the top of the arrays.

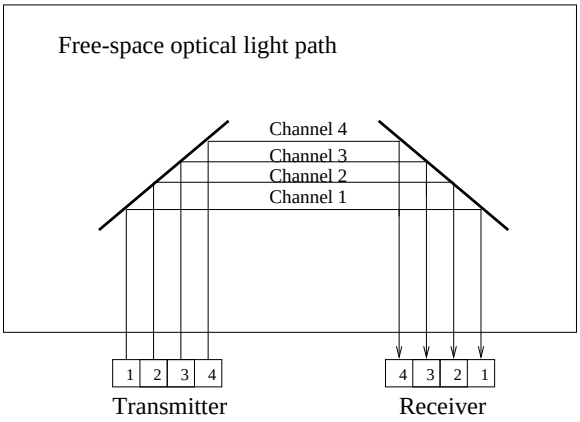

Figure 2.3a: Rigid free-space optical link

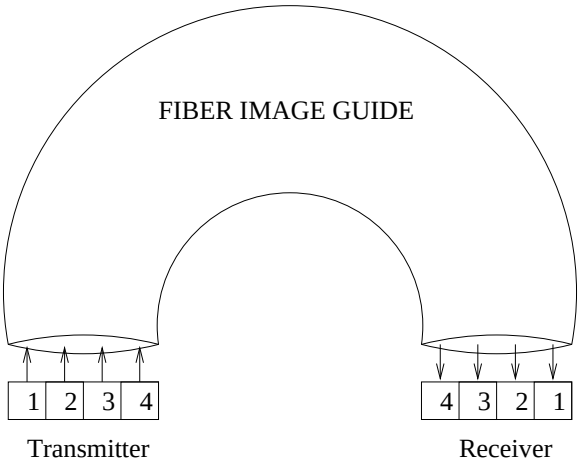

Figure 2.3b: Fiber image guide optical link

Use of free-space optics constrains the fan-in and fan-out for a cost effective operation. It is desirable to have a fan-in and fan-out of one under such conditions. This limitation of the technology points to a ring (Figure 2.4) as a reasonable topology. An additional benefit of a ring topology is that standard cache coherence mechanisms will function properly provided these transactions propagate all the way around the ring.

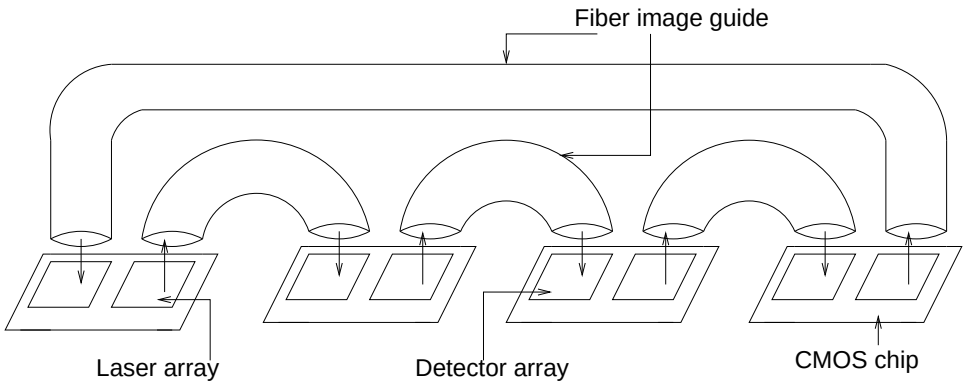

Figure 2.4: A four chip optical ring topology

2.2 System Descriptions

2.2.1 Network Architecture of the Multiring

This subsection describes a multiring-based optical interconnect architecture proposed in [11]. The multiring architecture is an enhancement over the conventional ring architecture [32]. Figure 2.5 illustrates a 4 node multiring architecture, where each node has a subring dedicated to traffic destined for it. For example subring 4 has nodes 1, 2, 3 feeding data into the channel and all this data is being received at node 4. Messages hop from one node to another, before finally reaching the terminal node for that subring, much like in a daisy chain. The multiring organization reduces the addressing overhead by explicitly reserving a channel for each destination.

The number of VCSEL-detector pairs used for a particular channel sets the bandwidth allocated for the channel. Figure 2.6 illustrates the allocation of VCSELs and detectors for a four channel system with 16×16 arrays of optical elements. Here, the elements are divided uniformly between the channels, using space division multiplexing. Assuming a data rate of 1 Gb/s for individual elements, this configuration yields $16^2/4 = 64$ Gb/s for each channel.

To send a message, a node uses the VCSELs dedicated for the subring corresponding to the destination node. The data incident on the detectors is received by a node if it is part of the channel dedicated for it, otherwise they are transmitted to the next hop in the same channel they came from. For example, in a message transfer from node 3 to node 1 (refer Figure 2.7), node 3 transmits data on channel 1 using the lasers dedicated for that channel. This is then received by the detectors dedicated for channel 1 on node 4, as these are not dedicated for channel 4 the node does not

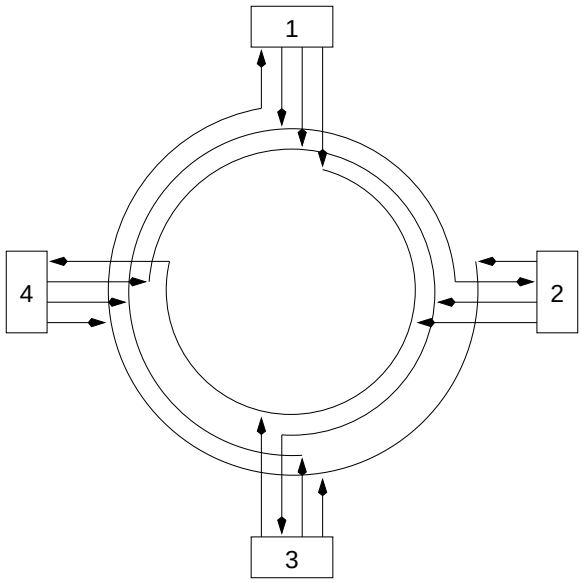

Figure 2.5: Multiring Topology - Showing logically separate channels

process the message, but transmits it to the next node in the ring (node 1) using the lasers dedicated to the channel on which the message was received. Node 1 receives these messages on the detectors dedicated for itself on channel 1 thus processes the message, without further transmission.

The multiring topology has the following advantages:

- Ideally Suited for Free Space Optical Interconnection: The optical fan-in and fan-out of each node is one. Single-hop communication is only with the two nearest neighbors.
- No Need for Explicit Destination Address Specification: An incoming message landing on the detectors assigned to channel i on node i's receiver automatically indicates that the message destination is node i.
- No Need for Explicit Routing: Since each channel is associated with a single receiver node, there is no complex routing necessary. If the node receiving the message is not the destination node, only a fixed forwarding operation is performed.

2.2.2 Media Access Protocol

A message from a given source to destination is broken down into smaller units called cells. Figure 2.8 illustrates an individual channel (channel 4) in a four node

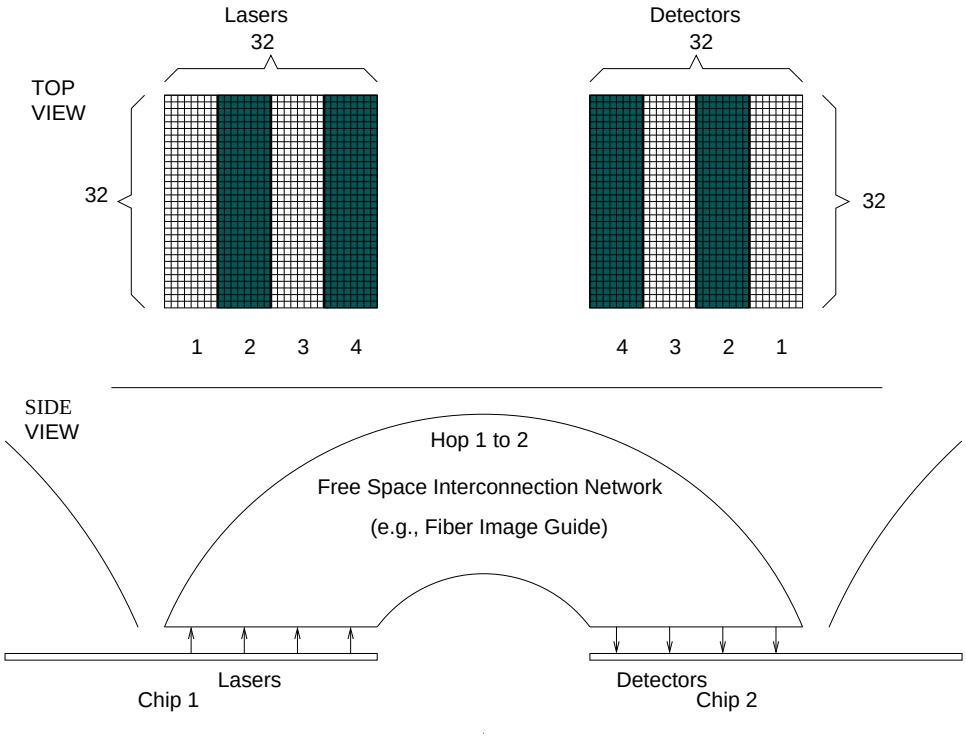

Figure 2.6: Allocation of VCSEL-detector pairs to a four channel system

multiring. In the basic design, channel priority is given to the upstream nodes, using the multiplexers in the network interface [14]. A downstream node does not transmit more than a single cell if it sees upstream traffic in the same channel (i.e, priority to upstream nodes at cell boundaries) [15]. This implies that the amount of buffering required at the intermediate nodes between the source and the destination is just one cell. Though not analyzed here, the scheme also enables per cell error correction [14].

Within a subring there will be cases when more than a single source compete for access to the channel. A Deficit Round Robin (DRR) mechanism is used to arbitrate in such cases.

2.2.3 DRR Scheduling

Deficit Round Robin Scheduling is used in the multiring to provide fair service to the various flows within the same subring. In other words, the distribution of the available bandwidth between the various sources in a subring can be decided by setting certain parameters inside the DRR protocol.

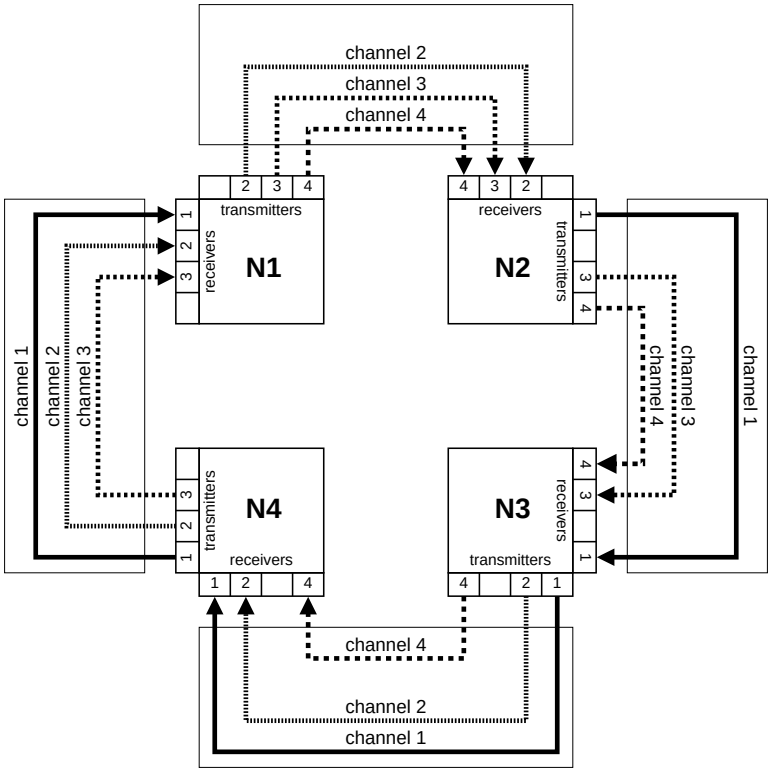

Figure 2.7: Conceptual diagram of a 4 node multiring

DRR scheduling was introduced in [46] for use in internet switches and routers, where the contention is for an output link. The associated overhead and achievable fairness is also discussed in [46]. It was modified in [15] for use in a Banyan topology interconnection network, and is described in [28] for use in a multiring. The DRR scheduler has the following attractive properties [11]:

- Flexibility. Nodes can be given different amounts of access to a channel by tuning parameters built into the protocol.
- Fast Decision Making. The DRR algorithm is fast since it needs to only examine the node in question to decide whether it should be given access to the channel.
- Fairness. DRR has been proven fair to the following extent: at any time, for equal priority channels, the difference in the amount of access granted to the most advantaged contender and the most disadvantaged contender is no more than three times the maximum message size.

In the original development of the DRR scheduler, all the information necessary to make scheduling decisions was present at a common location. Since the multiring

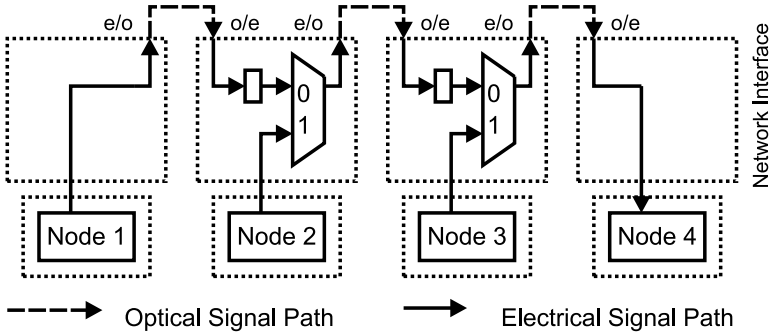

Figure 2.8: Channel 4 of a 4 node ring

is characterized by spatially separated channels, this mechanism must be adapted to work in this environment. Since every channel is associated with a particular destination, we assign the DRR scheduler for each channel, i.e., each DRR has an associated destination. Prior to sending a message on channel j (i.e., the destination node of the channel being j), a node i sends a control signal to node j requesting access to the channel. When the DRR scheduling algorithm (executing on node j) decides that sender i should have access, it replies with a control signal to i granting access to the channel.

The DRR scheduling algorithm, executing at each destination, maintains $N-1$ deficit counters, one for each potential message source. Each source node i is also assigned a quota q_i, indicating its relative bandwidth assignment on the channel. If all the quotas are equal, $q_i = q, \forall i$, the scheduler is to give equal access to the channel to all source nodes. The DRR module is present at both at the source and the destination nodes within a subring, as illustrated in Figure 2.9.

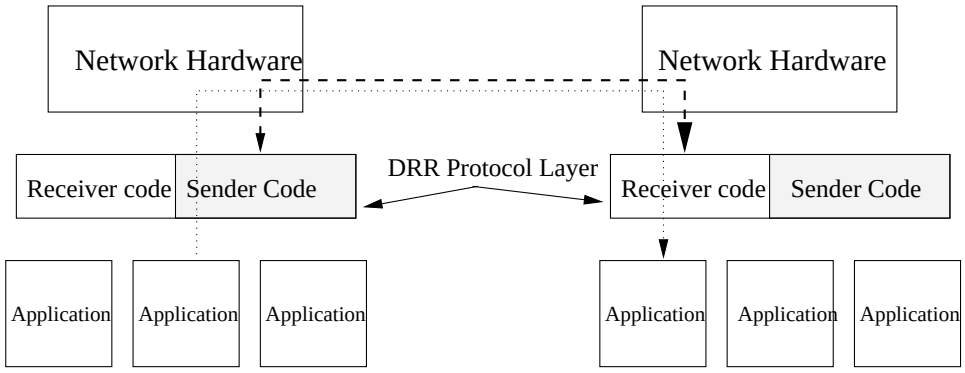

Figure 2.9: Placement of DRR fairness layer

The control messages from the sourceDRR to the destinationDRR indicate the size of the message that each source wants to send to the destination. The receiver

code of the destinationDRR has the responsibility of allocating the channel to a particular node. Upon receipt of a control signal from node i requesting access, the scheduler compares the size of the request to node i's deficit counter. If the request is to be granted (the message size is less than the deficit counter), a grant control signal is sent to node i and i's deficit counter is reduced by the size of the message. If the request is not granted (the message size is greater than the deficit counter), the control message remains in a request queue and is reconsidered in the next round.

Once per round, the deficit counters associated with each source are increased by their quota q_i. A round is defined as a period during which each source node contending for access is given the total allowed access as defined by its deficit counter. That is, a round is complete when every source is either not contending for access to the channel or has a deficit counter less than the size of its pending message. Details of the DRR scheduler are given in [29], and a complete description of its adaption to the multiring topology, including the implementation of in-band control signal delivery, is presented in [28]. The algorithm that determines which source gets access to the channel is summarized below :

Destination DRR Channel Allocation Process

```
WHILE (TRUE)
    IF (ActiveQueues > 0)
        DeficitCounter_i := DeficitCounter_i + Quota_i; {For active Queue i}
        WHILE (DeficitCounter_i > 0) AND (Queue_i) is not Empty)
            IF (MessageSize_i < DeficitCounter_i)
                ASK Source_i TO Send Message;
                DeficitCounter_i := DeficitCounter_i - MessageSize_i;
            ELSE BREAK; {Else proceed to other Queues}
            END IF;
        END WHILE;
        IF (Queue_i isEmpty)
            DeficitCounter_i := 0;
        END IF;
    END IF;
END WHILE;
```

The important characteristic to be noted here is that a source will continue to get access to channel as long as the size of the message to be sent by the source is smaller than the deficit of that particular source.

The bandwidth associated with each of the sources is thus dependent on the quota associated with it. A subring can be made fair by setting all the sources to have the same quota. As derived from the algorithm running at the destination DRR, the quotas give only the relative bandwidth allocated to the sources and do not set absolute bandwidth. It can thus be easily seen that if a source does not have any message to send, then the bandwidth which was given to that source by virtue of its quota will be distributed among the other sources.

2.3 Performance Characteristics of the Multiring

The system being analyzed here is the one described in [11]. A 8 node multiring system is modeled. Each node on the multiring has a SPA of size 32x32 with a 4x4 block of pixels used to convey a single bit. With the bandwidth obtained using a single VCSEL-detector combination being estimated at 1 Gb/s (Gigabits per second), the system on a whole is capable of delivering a net bandwidth of 64 Gb/s. A VLSI chip has been built at McGill University with a 256 channel, bi-directional optical interconnect [39]. The technology gave operational speeds of 400 Mb/s, which also shows that the speed estimates mentioned earlier are reasonable.

2.4 rICNS : A MODSIM III based Simulator for the Multiring

This section describes the simulation program written in MODSIM III to simulate the operation of the multiring. The program originally developed by Ch'ng Shi Baw [13] has been enhanced by Abhijit Mahajan [28] and the author. The multiring simulator has its origin in ICNS, a simulation framework designed to ease the development of simulation models for optically interconnected systems [8]. At a very high level, an interconnection network can be abstracted as a system composed of terminals that generate and consume messages, and links and switches that facilitate the transportation of messages from one terminal to another. The design of ICNS is not limited to photonics in the processor-to-processor interconnection network, it is used to model both multicomputer systems and switching fabrics for internet routers.

ICNS has been used to model a pair of systems. The first is the Gemini interconnect, a parallel photonic and electronic network that utilizes lithium niobate optical switches to construct a circuit-switched high-bandwidth data path in the switching fabric. The second is a photonic multiring interconnect, in which 2-D arrays of Vertical Cavity Surface Emitting Lasers (VCSELs) and photodetectors are used to provide high-bandwidth I/O to/from CMOS chips. The variety in photonic technologies used, as well as the distinct architectures that result, point to the flexibility of the ICNS framework.

In the system architecture described earlier, the multiring is currently being simulated as a set of independent subrings being driven by 8 sources that each deliver data to 7 subrings. There are a total of 8 subrings, and the sources are modeled in a way that a source does not send messages to the subring where its node is the destination node on the ring.

2.4.1 Simulation Model for the Multiring

As mentioned earlier the simulator was written in MODSIM III, a powerful and versatile object oriented language for discrete-event simulation. MODSIM III was developed by CACI Products Company and is now managed by Compuware. For details about the language, the reader is refered to the MODSIM III software manuals, tutorials and user guides [6].

The multiring is modeled as a set of independent subrings which are driven by a common generator (source model). Figure 2.10 shows an overall model of the system, with the majority of the blocks for a 4 node case. The message sender block (Node 1) is responsible for sending the messages destined for node 4 (generated by the global generator at node 1), using channel 4.

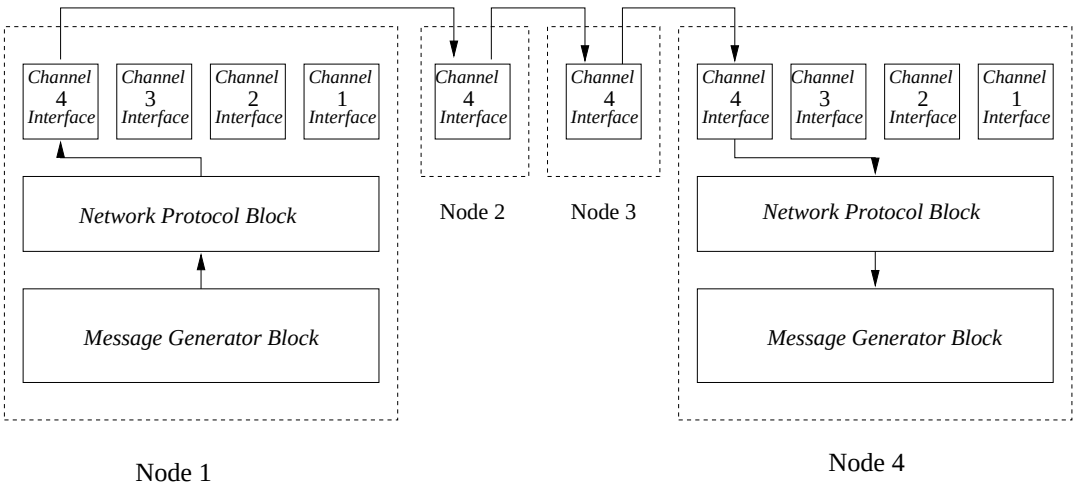

Figure 2.10: Model of the ring used in simulation

Each of the subrings can be seen as being composed of layers (as in networking) through which the information traverses. The main layers in this model are the Application, Network and the Data Link layers.

The Application layer is responsible for generating messages to the various destinations (Message Generating Block). The simulation has been written so as a large number of traffic pattern distributions can be simulated. The messages generated by the generator module are divided up into cells by the network protocol block. At each node there also exists a Ring Channel Interface, which deals with the issue of media access protocol.

The Network layer routes the cells from the source to destination proceeding in hops between the intermediate nodes. The design of the multiring ensures that the message stays on the subring (channel) dedicated to its destination during the entire duration of the transmission. The time taken by the each cell for each of its hops is determined by the cell length and the bandwidth allocated to that particular subring (channel).

Table 2.1 provides descriptions of some important objects which form the core of the simulator.

Table 2.1: Brief description of key objects in the simulator

Object	**Purpose**
rGlobalGenObj	Generates Messages of some specified randomly distributed length and destinations.
rMessageObj	It is the initial object that is created by the rGlobalGenObj
rCellObj	A Message is divided into cells of fixed size which are represented by this object
rTerminalObj	Divides the message into cells and delivers it to the appropriate subring
rCIObj	Each node has a CI (Channel Interface) connected to it for each of the subrings. It transfers the cells to the destinations, i.e, from one CI to another in hops
rDRRModule	It models the DRR scheduling scheme for implementing the desired "fairness" within a subring.
NetworkObj	It models one of the subrings with the Multiring
rMultiringObj	It models the entire Multiring and is essentially a container for all the NetworkObjs

Chapter 3

System Reconfiguration

3.1 Introduction

This chapter describes the general desire for system reconfiguration. Section 3.3 provides insight into the issue of *fairness* in a system and how it relates to the system described in chapter 2. It goes on to describe the idea of reconfigurability in relation to the multiring architecture. Sections 3.3.1 and 3.3.2 describe the physical techniques that are used to change the bandwidth allocated to a source-destination pair in the multiring architecture. Some of the material in this chapter comes from from [10].

There are many applications where there is *a priori* knowledge of the communication pattern and the bandwidth requirements. This knowledge can be used to reconfigure the interconnection bandwidth when there is a change in the requirements. The reconfiguration is performed when needed and is not done otherwise. This is what is termed as *Static Reconfiguration* and is typical of signal processing applications which have alternating communication and computation phases as the application progresses. Reconfiguration here is done before the start of the communication phases and remains static during that phase phase. More details on this is presented in Section 3.4.

Another type of reconfiguration is *Dynamic Reconfiguration*, where we consider the set of applications in which the demand on the interconnection network cannot be predetermined. The interconnection network in this case is reconfigured at regular (or possibly irregular) intervals with an attempt to satisfy the needs of the system at that particular time. Internet switching systems are classic examples where the load on any particular flow cannot be determined ahead of time. More details on this are presented in Section 3.5.

This chapter also describes the simulation models used to analyze both static and dynamic reconfiguration techniques. Section 3.4.1 provides the details and assumptions made

for static reconfiguration and Section 3.5.1 describes the same for the dynamic reconfiguration case. Issues with determining steady state conditions in heavy tailed distributions are also discussed here, including our approach to dealing with the problem.

3.2 Motivation for Reconfiguration

Reconfigurability in the system described here refers to the ability to change certain parameters that characterize the system at execution time. The parameter of interest in our case is the bandwidth capacity of the system on a per flow basis. Understanding the implications obtained by changing the interconnect bandwidth to match the needs of the application is the primary motivation for this work. In particular, this work discusses the benefits obtained in the areas of parallel computation and network switching.

3.2.1 Parallel Computation

Design of a parallel application usually consists of dividing the completed task into several processes that can be executed in parallel and finding a hardware platform that offers acceptable performance [2]. Many applications that are executed in parallel are both computationally intensive and involve comparable interprocessor communication.

Though applications are often coded to utilize all of the available resources, it often happens that many applications have varying demands on the interconnection network. Parallelized applications often can be characterized by a sequence of alternating communication and computation phases, and communication phases themselves do not have the same communication pattern or volume of traffic every time. Studies that attempt to characterize scientific applications [33, 34] show that most data-parallel applications present a behavior which is cyclic with time. Signal processing applications fall into this class of applications. Modern parallel machines in turn run many such applications, which further increases the variation in the demand on the interconnection network. In such scenarios it is clearly beneficial to have a interconnection network that can be reconfigured based on the needs of the application that is being executed on the multicomputer system.

3.2.2 Internet switching

Besides the class of applications mentioned earlier, there are some which do not have any regular or cyclic characteristics i.e., their bandwidth cannot be predetermined. These applications are often characterized by bursty message traffic, such as in real time video applications. We explore the possibility of obtaining a speedup in these cases by dynamically reconfiguring the bandwidth allocations of the interconnect during execution time.

In this thesis, our focus is observing the benefits in extending this reconfigurable multiring architecture to network switching. We discuss the advantages of using this technology as a switch fabric in network routers.

The transmission capacity of optical links has increased considerably over the last few years, in some cases outstripping the ability of electronic switches to keep up, which can in turn lead to long input queues. The nature of this build up is also unpredictable because of the bursty nature of the traffic. This is an ideal case to see the benefits of having a reconfigurable switch fabric.

3.3 Physical Techniques for Reconfiguration

Reconfigurability in this system refers to the allocation of bandwidth resources based on the requirements of the application or the network. As described earlier the multiring architecture consists of individual subrings, with each subring assigned to a given destination. To deliver a message to node i all the sources put their messages in subring i. This essentially makes the bandwidth allocation process in the multiring a two level hierarchy. The first level being allocation of the bandwidth across the various subrings, followed by distributing the allocated bandwidth to the various sources within a subring.

There are two levels of reconfiguration corresponding to the two levels of bandwidth allocation in the multiring. The first being Laser Channel Allocation (LCA), which reconfigures the bandwidth allocated to the various subrings and Deficit Round Robin (DRR) allocation, which changes the relative bandwidth allocated to the sources within a subring.

3.3.1 Reconfiguration among the subrings

Laser Channel Allocation (LCA) is the mechanism responsible for setting the amount of bandwidth allocated to a particular subring from the total available bandwidth. As shown in Figures 3.1a and 3.1b, the number of VCSEL-detector pairs allocated to a particular channel determines the amount of bandwidth allocated to the particular channel. Uniform allocation (which is the default configuration) shown in Figure 3.1a refers to the case when all the channels have the same number of VCSEL-detector pairs allocated to them, whereas in the reconfigured case (Figure 3.1b) channel 2 has twice the bandwidth compared to channel 1 and three times the bandwidth compared to channels 3 and 4.

Essentially the number of optical paths, and hence the bandwidth associated with each of the subrings, is modified by using the LCA mechanism. The bandwidth allocated to each of the subrings, once set, is fixed until another LCA reallocation is performed. The

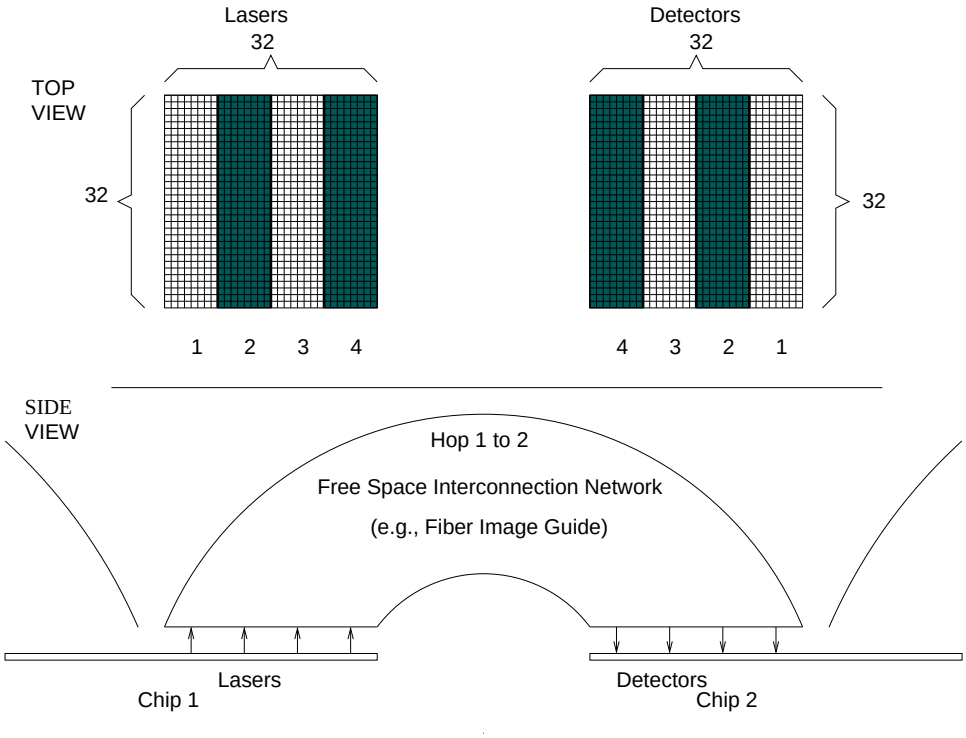

Figure 3.1a: Uniform allocation of VCSELs to channels

bandwidth allocated to a node by this method can be shared only by the sources within the subring. This is a rigid allocation of the bandwidth between the various subrings.

This mechanism is used to change the bandwidth allocated with a particular output port i.e., the number of optical paths dedicated for delivering messages to this output port is changed by this mechanism.

Figure 3.2 illustrates the low level mechanisms on a node which permit LCA reconfiguration. We see here that there are four inputs and four outputs from this node, which corresponds to optical data coming into the node and optical data going out of the node. After converting the data into the electronic domain, we can either route the data to the next hop or process the data on this node by using the demultiplexors at the input. We can also use the multiplexors shown in this figure to select what data actually goes out of the transmitters from the node.

The process of changing the multiplexors and the demultiplexors can be done in one clock cycle. It is the issue of synchronizing all the nodes to confirm to the same configuration which will take a longer time.

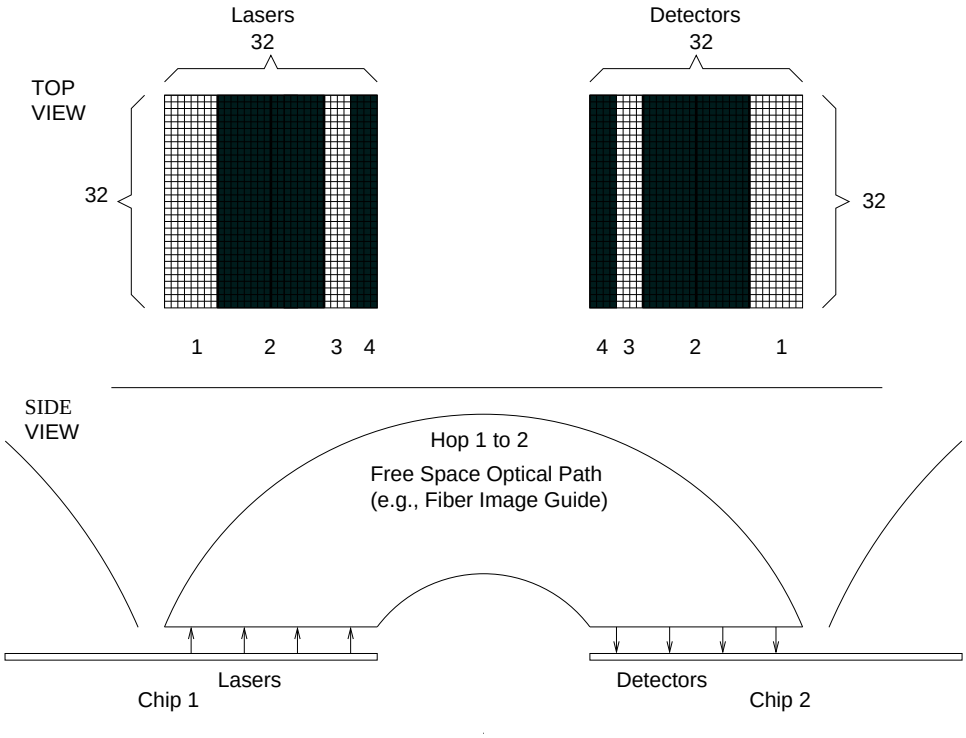

Figure 3.1b: Reconfigured allocation of VCSELs to channels

3.3.2 Reconfiguration within a subring

The Deficit Round Robin (DRR) scheduler described earlier gives flexibility in allocating bandwidth within a subring. In an $N \times N$ system each subring has $N - 1$ source nodes and a destination node. Each flow within a subring can be allocated a share of the total bandwidth allocated to that particular subring, simply by altering the quota associated with that flow.

As described in Chapter 2, Section 2.2.3 the DRR algorithm running on the destination node maintains a quota q_i and a deficit count for each of the source nodes. The algorithm describes the way in which the access of the channel to the source is given within a subring. The essence of the DRR scheduling algorithm is that a source can send its message in the subring only when the deficit count associated with it is more than the size of the message it wants to send, if this criteria is not met then the deficit count of that particular source is incremented by the fixed quota associated with that source at the end of a round. The scheduler checks the criteria for all the sources in a round robin process. The effect of this is that sources that have a larger quota receive a proportionally larger fraction of the available capacity.

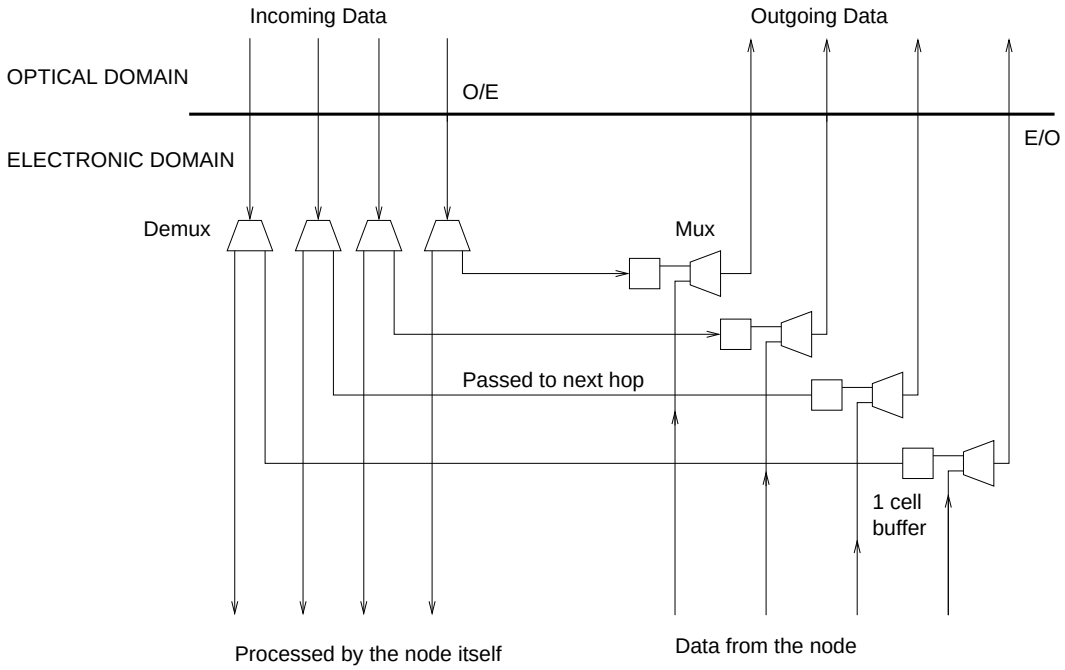

Figure 3.2: LCA reconfiguration at each node

In the case that a particular source does not have any traffic to send, then the bandwidth allocated to that source will be distributed among the other nodes. The DRR algorithm only allocates the relative bandwidths between the various sources, the amount of traffic a source has to offer also determines the amount of bandwidth actually allocated to that particular source. The relative bandwidths can be modified by changing the quota associated with the sources, the higher the quota the higher is the share of the bandwidth.

3.3.3 Summary of Physical Mechanisms

The above two mechanisms for reconfiguration viz., Laser Channel Allocation (LCA) and Deficit Round Robin (DRR), can be used to control the allocation of the resource on a per-flow basis. LCA determines the absolute bandwidth to a destination, where as DRR determines the relative bandwidth of the sources within a subring of a multiring. Chapter 4 further discusses the implications of these reconfiguration techniques and gives the benefits obtained by these techniques in the scenarios discussed.

3.4 Static Reconfiguration - Description

As mentioned earlier, many applications run on parallel multicomputers are characterized by a sequence of alternating computation and communication phases. Also, the bandwidth

requirement on the interconnect during the communication phases is known *a priori* in many of these cases. For the set of such applications we propose the scheme of *Static Reconfiguration* where the bandwidth allocation for the various flows in the interconnect is set before the start of a communication phase. This mechanism of reallocating the bandwidth is termed *static* as the allocation is static for the duration of the communication phase.

Typical signal processing applications are characterized by these alternating communication and computation phases. Within a single application there can be different types of communication patterns, such as all-to-all, broadcast, point-to-point, gather, etc. The number of nodes that participate in a communication phase can also vary. These change the requirements on the interconnect, demanding varying bandwidth for different flows across individual phases.

The demand in these cases is known *a priori* for applications like synthetic aperture radar, beamforming, etc. The demand is known on a per flow basis and can be used to set the parameters discussed earlier to meet the needs of the application. The control that changes the bandwidth associated with each flow of the interconnect is known at compile time for each application.

3.4.1 Simulation Model Details

For the simulation of static reconfiguration, two sets of applications are considered. The first is a pair of real applications where, from an understanding of the application, the communication patterns are known. The second set consists of synthetic applications whose properties have been chosen randomly from a set of common communications patterns.

The two real applications include synthetic aperture radar (SAR) image formation, and beamforming (BF). The SAR application, for example, can be viewed in terms of the phases shown in Figure 3.3. For this application, the first communication phase consists of data being input from the sensor array (a broadcast). The first computation phase consists of range processing. The second communication phase is a corner turn operation (an all-to-all pattern). The second computation phase is azimuth processing, and the final communication phase is the output of formulated SAR images (a reduction).

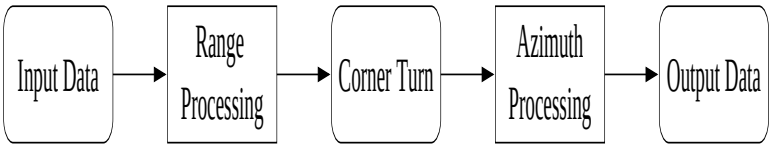

Figure 3.3: SAR phases.

The communication patterns associated with each SAR phase are shown in Figure 3.4. Nodes 1 and 8 correspond to the input and output nodes respectively. Nodes 2 through 7 correspond to processor nodes which perform the computation. Thus, in phase 1 the communication pattern corresponds to distributing the input data from node 1 to processing nodes 2 through 7. In phase 2, an all-to-all exchange of data between the processing nodes takes place, while in phase 3 a reduce operation occurs which aggregates the final image from the processor nodes to the output node 8.

Similarly, the properties of the BF application have been determined and modeled. The beam forming application consists of 5 communication phases viz., Broadcast from the sensor node, followed by an all-to-all between the 6 computation nodes. These two stages are similar to that in SAR. These are followed by a gather operation at node 7 in which all the other compute nodes send a part of their processed data. This is then followed by a broadcast of the data from node 7 to the other computation nodes. This is followed by a gather operation at node 8 where all the computing nodes (2 to 7) send data.

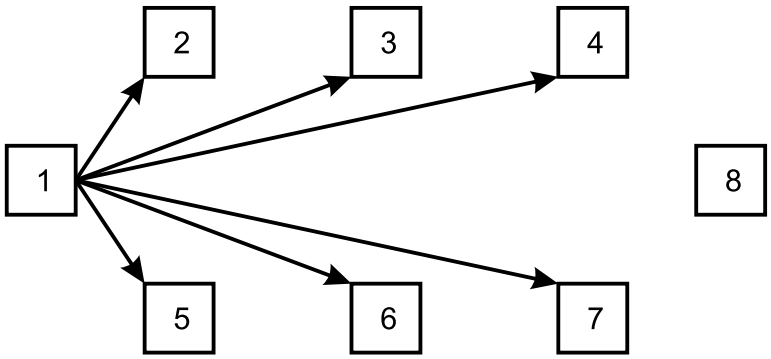

Figure 3.4a: Broadcast from sensor node to compute nodes.

Ten synthetic applications were also analyzed. For each, the following three application parameters were generated in a random fashion:

- **Number of Phases:** The number of communication phases was selected uniformly between 3 and 6.
- **Communications Pattern:** Four communications patterns commonly associated with space-time adaptive algorithms were considered with equal probability:
  - **All-to-All:** All the nodes exchange data with each other.
  - **Broadcast:** One randomly selected node sends information to a random selection of other nodes.
  - **Reduce:** A random selection of nodes sends information to a single randomly selected destination.

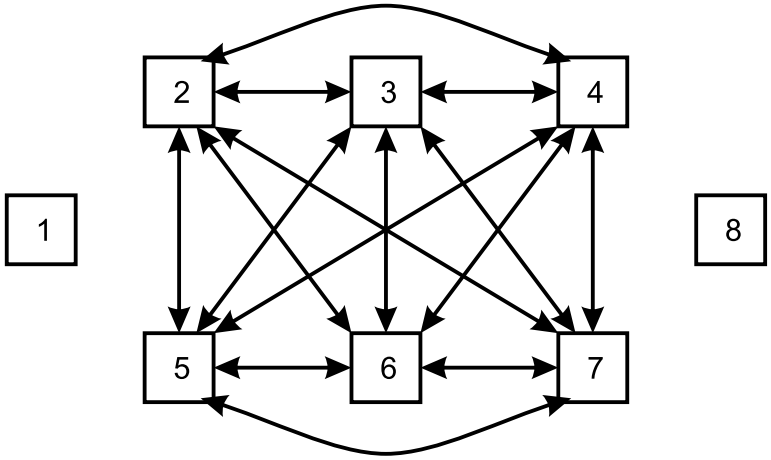

Figure 3.4b: Corner turn between compute nodes.

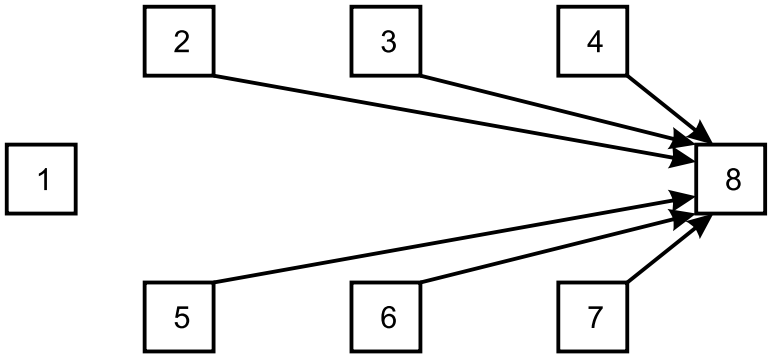

Figure 3.4c: Reduction from compute nodes to output node.

  - **Point-to-Point:** A random set of source-destination pairs are selected with communication being required between the pairs.

- **Communication Volume:** For each flow associated with each pattern, the amount of information to be transferred was randomly selected from a fixed set of message sizes that spanned two orders of magnitude.

Details about these parameters for all the synthetic applications are given in Table 3.1. The entries in the table show communication characteristics of the synthetic application. The table shows the communication pattern, participating nodes and the communication volume for all the communication phases of the synthetic applications.

The interconnect performance for these applications with all possible combinations of the two reconfiguration techniques (DRR, LCA) are studied. The performance of the interconnect with reconfiguration is compared against a control case of uniform allocation, in which all the flows have equal bandwidth allocated to them.

Table 3.1: Details of the synthetic applications simulated

	Phase	Pattern	Source(s) / Destination(s)	Message Size Ratio(s)
App A				
	1	P2P	S : 1 3 4 6 8	0.1 0.1 0.3 0.3 0.3
			D: 4 5 6 8	
	2	Broadcast	S: 8	0.3
			D: 2 4 5 6 8	
	3	A2A	S: 1 2 4 5 6 7	0.3
			D=S	
App B				
	1	P2P	S : 4 5 7	0.03 0.3 0.3
			D: 1 3 4	
	2	Broadcast	S: 6	1.0
			D: 2 3 5 6 7	
	3	Broadcast	S: 3	0.03
			D: 3 4 8	
	4	Broadcast	S: 8	0.1
			D: 3 7 8	
	5	P2P	S : 3 6	0.1 0.1
			D: 4 5 8	
App C				
	1	P2P	S: 4 7 8	0.1 0.1 0.3
			D: 3 7 8	
	2	Broadcast	S: 5	0.1
			D:3 5 6	
	3	P2P	S: 4 8	0.03 1.0
			D: 1 2 4 6 8	
	4	Reduce	S: 6 3	1.0
			D: 3	

Table 3.2: Details of the synthetic applications simulated contd.

	Phase	Pattern	Source(s) / Destination(s)	Message Size Ratio(s)
App D				
	1	P2P	S: 2 5 6 7	0.3 0.03 1.0 0.1
			D: 1 4 7	
	2	Broadcast	S: 7	1.0
			D: 1 5 6 7 8	
	3	A2A	S: 2 3 4 6 8	0.1
			D=S	
	4	Reduce	S: 1 2 3 6 7 8	1.0
			D: 7	
	5	P2P	S: 2 3 4 5 7 8	0.1 0.03 0.03
			D: 1 3 4 6	0.03 1.0 0.03
App E				
	1	P2P	S: 2 4 6 7 8	0.03 0.3 0.1 0.1 0.1
			D: 3 4 6	
	2	A2A	S: 1 3 5 7 8	1.0
			D=S	
	3	Reduce	S: 2 4 5 6	0.1
			D: 6	
	4	Broadcast	S: 6	0.3
			D: 1 2 3 6 8	
	5	A2A	S: 1 2 8	0.03
			D=S	
	6	Reduce	S: 1 5 6 7	1.0
			D: 5	
App F				
	1	Reduce	S: 2 5 8	1.0
			D: 5	
	2	Reduce	S: 1 3 6 8	0.3
			D: 6	
	3	A2A	S: 3 5 7	0.1
			D=S	

Table 3.3: Details of the synthetic applications simulated contd.

	Phase	Pattern	Source(s) / Destination(s)	Message Size Ratio(s)
App G				
	1	P2P	S: 3 6 8	0.3 1.0 0.1
			D: 1 4 7	
	2	Reduce	S: 2 5 7	0.1
			D: 5	
	3	Reduce	S: 1 2 3 5 7	1.0
			D: 1	
	4	Broadcast	S: 8	0.03
			D: 2 3 4 5 8	
App H				
	1	Reduce	S: 2 3 6 7 8	0.3
			D: 2	
	2	P2P	S: 1 3 6	0.3 0.03 0.03
			D: 1 2 7 8	
	3	A2A	S: 2 3 8	0.3
			D=S	
	4	Reduce	S: 1 5 6 7 8	1.0
			D: 7	
App I				
	1	P2P	S: 1 2 3 4 7	0.3 0.03 0.1 0.1 0.3
			D: 57	
	2	Reduce	S: 3 4 8	0.3
			D: 4	
	3	Broadcast	S: 1	0.1
			D: 1 2 3 4 7	
	4	P2P	S: 1 7	0.3 1.0
			D: 1 4	
AppJ				
	1	P2P	S: 1 2 8	0.3 0.3 0.03
			D: 1 4 7	
	2	A2A	S: 3 5 7	0.03
			D=S	
	3	P2P	S: 1 2 4 5	0.03 1.0 0.03 0.1
			D: 1 3 5 8	
	4	Broadcast	S: 6	0.03
			D: 3 6 7 8	
	5	Reduce	S: 1 2 3 5 6	1.0
			D: 2	

The following is the set of experiments that were conducted for the purpose of performance evaluation:

- **UA - Uniform Allocation:** Over all phases, the bandwidth was divided evenly among the rings and, within each ring, sources were given equal quota. This ensures uniform allocation over all source-destination pairs and represents the base case where no reconfiguration is done.
- **DRR - Deficit Round Robin:** Available bandwidth is evenly divided among the rings. Within a ring, knowing the bandwidth requirements of each source-destination pair (or flow), the quota associated with pairs in the ring are adjusted to reflect the application flow bandwidth demands. This is done at the start of each phase and represents ring-level reconfiguration.
- **LCA - Laser Channel Allocation:** Knowing the bandwidth requirements of each source-destination pair (or flow) one can determine the bandwidth requirements associated with each ring. Based on this, LCA divides up the total bandwidth available to reflect the bandwidth needs of each ring. This is done at the start of each phase. Within each ring, the quota associated with each flow are set equal.
- **DRR-LCA - DRR and LCA together:** Knowing the bandwidth requirements of each source-destination pair, both DRR quota and LCA ring bandwidth allocations are performed at the start of each phase.

The input traffic was generated as a single burst of messages size. The message from one node to another was divided up into smaller units (cells) and was queued up at the source. The source DRR module at each source in the subring queues up the cells and the destination DRR module of the terminal node dequeues these cells based on the DRR scheduling algorithm.

When simulating DRR reconfiguration we change the quota q_i of the sources within a particular subring. This enables us to set the desired relative bandwidth between sources within a subring. For the uniform allocation case the quota associated with all the sources is the same, which means that all the sources have equal access to their subring. For the cases where we simulated the DRR reconfiguration, as we know the requirements ahead of time, the quota associated with each source was changed before the start of the communication phase.

Simulation of the LCA reconfiguration technique involved altering the number of cells that were delivered using a particular subring, according to the bandwidth that was allocated to the subring theoretical. For example, consider the case in which the uniform allocation of VCSELs a particular subring had N cells of traffic, and when reconfigured it

gets *twice* as many VCSELs. The same in simulation would amount to a traffic of $N/2$ cells in that particular subring. The number of cells would correspondingly increase for the case that the number of VCSELs allocated for a particular subring decreases. As the bandwidth requirements in the *static* reallocation case are known before compilation time, we change the number of cells that are actually being sent to simulate the reallocation of bandwidth.

3.5 Dynamic Reconfiguration - Description

In contrast to *static reconfiguration*, with dynamic reconfiguration there is no *a priori* knowledge of the communication bandwidth requirements, nor is there is a specific pattern of bandwidth requirements. The purpose of dynamic reconfigurations in these cases is to react to the instantaneous load that is experienced.

In conventional switches the allocation of bandwidth to the various ports is fairly rigid. In cases where there is a burst detected on one of the ports, the switch in most cases is not able to react to it appropriately. We propose a reconfiguration mechanism by which we can change the bandwidth allocation between the ports of a switch, to improve the performance in the case of these unbalanced load situations.

The dynamic reconfiguration is of interest in cases where there is an uncertainty in the amount of traffic and its pattern. For example, in case of a network switch input queues give an estimate of the bandwidth required for each of the flows. The bandwidth requirement for this case does not follow any predictable pattern. The unpredictability of this system makes it an ideal candidate for dynamic reconfiguration.

The control that determines the bandwidth configuration of the system has inputs that vary periodically (i.e., varying bandwidth requests). A control algorithm is used to determine the per flow bandwidth assignment. This algorithm is applied synchronously to all the source and destination pairs to keep the reconfiguration overhead to minimum. Some candidates that can be inputs to the control algorithm are:

- **Determinable Patterns**: In these kind of loads where there is a *a priori* knowledge of the traffic pattern (e.g., due to bandwidth reservations), we can reconfigure the switch on an as needed basis. This is typical of circuit switching, where bandwidth is reserved for particular flows for sometime and then distributed to other flows. Though this model is comparably easy to implement, the traffic model expected for this kind of reconfiguration is not common.

- **Instantaneous Queue Lengths**: In this case we reconfigure the per flow bandwidth based on the instantaneous queue lengths at the various nodes at periodic intervals. This is a straight forward way of reconfiguring the switch, where we don't need to

maintain any significant state information. For bursty traffic in the network, this method works well only if the network is monitored and reconfigured frequently.

- **Quality of Service**: This case corresponds to the case when some flows have higher priority compared to others. If the load on any of these flows is higher than the allocated resources, the interconnect can be reconfigured to meet the quality assurance required.

- **Filter-based Reconfiguration**: This is a periodic reconfiguration of the fabric, where decision are made based on the queue lengths after passing the inputs through a low pass filter, such that we are able to set a maximum threshold on the amount of bandwidth allocated.

- **Smaller the better**: This scheme gives priorities to flows that do not have huge backlogs to transmit. This helps the overall delay numbers, as in most cases the smaller message getting priority will not significantly affect the delay values for the larger message flows.

In this thesis we reconfigure the switch periodically based on instantaneous queue lengths at the input port. The bandwidth allocations to the various flows are proportional to the total messages queued for that flow. the mechanisms used for dynamic configuration are the same as for static reconfiguration. One way to perform dynamic reconfiguration is to perform the static reconfiguration techniques discussed earlier periodically. The LCA and DRR techniques for changing the bandwidth allocations within the multiring are used, with the amount of bandwidth allocated is based on runtime control algorithms.

We first divide the total bandwidth available in the system across the 8 channels, giving each an amount directly proportional to the backlog at each destination port using LCA. We also ensure that each subring is given a bandwidth of at least 1 GB/s. We then proceed to allocate relative bandwidth for the sources within each subring. This is done by changing the quantum associated with each source in the DRR scheduling algorithm. The quantum each source gets is proportional to its relative load within each subring. In this case also we give each flow a minimum quantum of 200 cells to prevent starvation of any source.

3.5.1 Simulation Model Details

To study the performance of dynamic reconfiguration, an 8 port network switch was simulated. The ports of switch were arranged in a multiring configuration (Figure 3.5). Data can be sent from any port to any other port. The multiring configuration is the same as

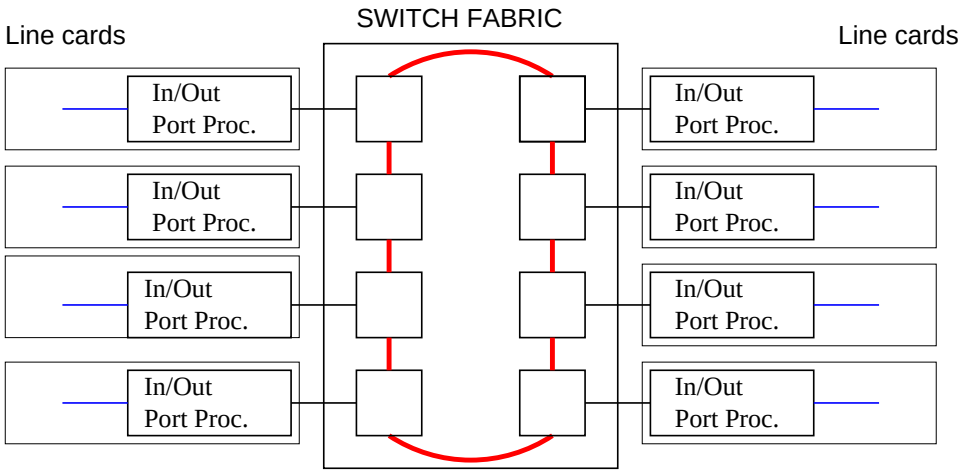

Figure 3.5: Multiring technology as part of a router switch

described in Chapter 2. Each port has an unique subring dedicated for traffic destined to it.

The input traffic for the switch is *self-similar*. Self-similar traffic is generated by a mechanism proposed in [37] representing an interactive transfer of files over a network, the size of the files being drawn from a Pareto distribution. The simulation uses a fairly heavy tail distribution for file sizes, and a exponential distribution for the interarrival time of the the message (file). The inter-arrival time for the exponential distribution is the OFF period of the ON/OFF model [47]. The mean file size of $\approx 4.1KB$ for the heavy tail is also obtained from [37].

The interarrival time being an exponential distribution in contrast to a Pareto distribution (heavy tailed) is consistent with the results in [47], which is further demonstrated in [37]. Simulations in [37] show that a heavy-tailed idle distribution is not needed, and a heavy-tailed file size distribution is by itself sufficient to produce self-similarity. The relationship between the file size distribution and the traffic self-similarity is not significantly affected by the changes in network resources, topology, traffic mixing, or the distribution of interarrival times [37].

In our simulations it is assumed that the time taken for applying the control algorithm for the various queues is constant. It is also assumed that no traffic is delivered from one node to another during the time that the control algorithm executes.

Steady State Effects

Due to the effect of the heavy tail distribution which drives the file sizes (message sizes) of the generators, steady state is not reached in these simulations. To measure the mean queue sizes, the simulation run was divided into 20 batches, with each batch corresponding

to an interval of 5×10^5 cell times, with queue sizes being measured every 512 cell times. Figure 3.6a shows the mean queue lengths during the various intervals within a particular subring (subring 3), and Figure 3.6b shows the mean queue lengths across the various destination, for a utilization of 100%. We see here that sources 5 and 6 dominate the characteristics in Figure 3.6a and all the other sources have mean queue sizes close to zero.

The batch means for utilization of 50% for the same cases is illustrated in Figures 3.7a and 3.7b. These plots show us that steady state is not reached for the duration of the simulation run, thus it would be inappropriate to compare the performance of the system just during the time that the messages are generated. It is important that we compare the same set of messages for the uniform allocation and reconfiguration cases.

For this purpose, generation of the messages is stopped after a predetermined duration (1×10^7) cell times. The messages queued up at the various sources are allowed to drain. The sources within a subring drain in the order imposed by the DRR scheduler. The performance characteristics presented in the next chapter are obtained over the entire simulation run, including the drain phase of the experiment.

The performance of the reconfigured interconnect is compared to the case in which bandwidth is allocated uniformly to all the source-destination flows in the network. All the reconfiguration runs of a set have the same input pattern (seed) as that of the uniform case, so that they correspond the same input traffic pattern. Various sets of these experiments are simulated with different seeds for the message generators, to get statistical confidence in the results obtained.

The next chapter presents and analyzes the performance of the interconnect with the various reconfiguration techniques discussed. It goes on to compare the various reconfiguration techniques highlighting the benefits in each case.

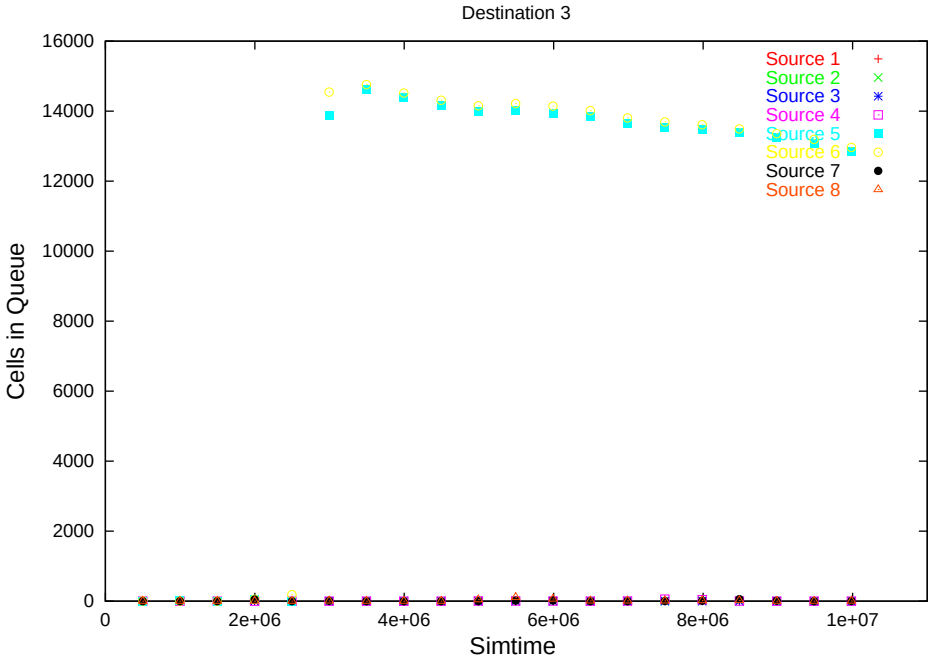

Figure 3.6a: Batch mean for 100% utilization - subring 3.

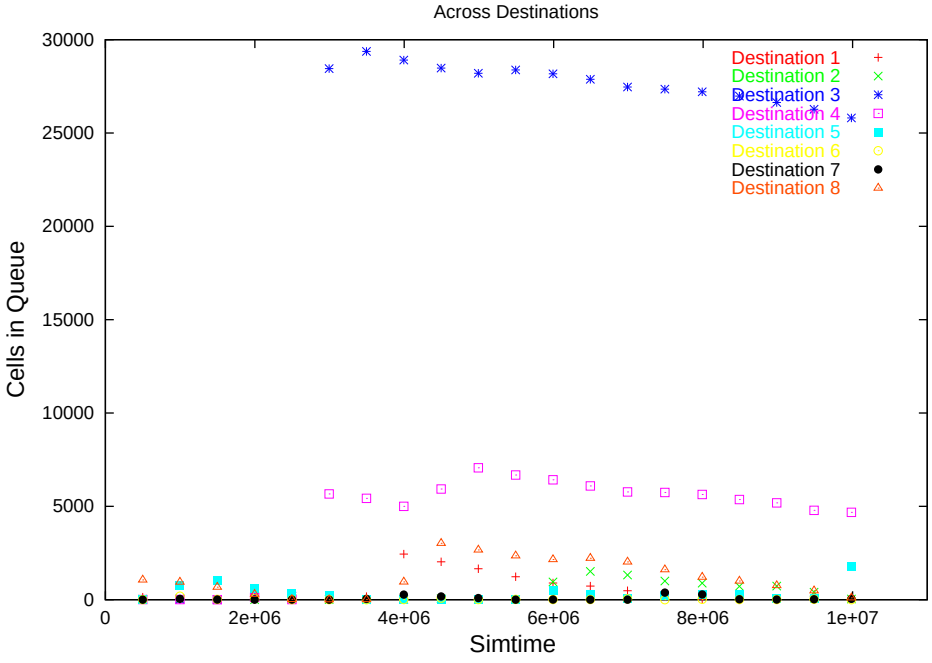

Figure 3.6b: Batch mean for 100% utilization - Across all sources for a given destination.

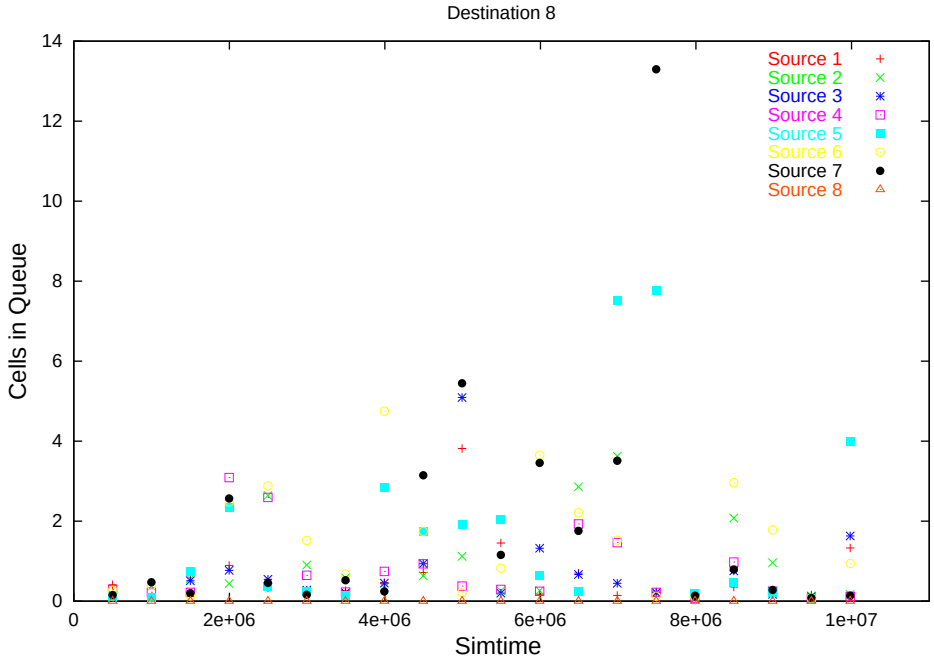

Figure 3.7a: Batch mean for 50% utilization - subring 3.

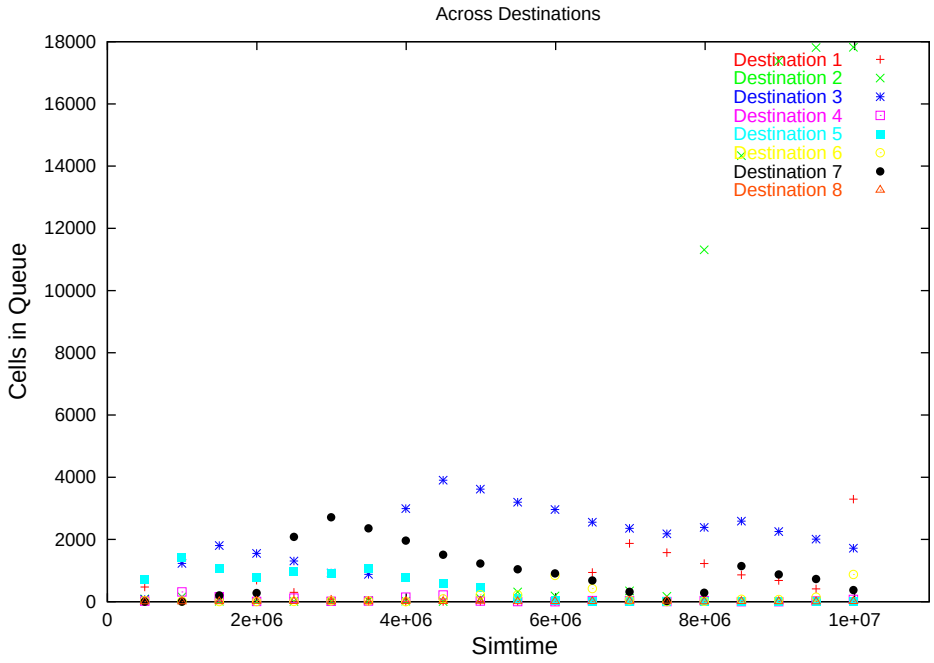

Figure 3.7b: Batch mean for 50% utilization - Across all sources for a given destination.

Chapter 4

Simulation Experiments and Results

4.1 Introduction

This chapter explores the performance implications of the reconfiguration techniques discussed in Chapter 3. The chapter provides simulation results for both static and dynamic reconfiguration, the static case being analyzed first followed by the dynamic case.

Section 4.2 discusses the applications simulated for obtaining the performance numbers for the static reconfiguration. Section 4.2.3 explains the various metrics used to evaluate the performance of the system, including the *latency fairness metric*, *speedup*, etc. It also provides the comparison between the metrics for the various experiments simulated under the static reconfiguration.

Section 4.3 presents the details about the various experiments simulated to evaluate the performance of the dynamic reconfiguration. It also gives details about the how the source model for driving the message generators was constructed, such that it represents the traffic pattern seen on the internet. Section 4.3.1 presents the results obtained from the various scenarios simulated. Similar to the static reconfiguration, we discuss the effectiveness of reconfiguration by considering parameters such as mean message delivery time and also the variability of message delay. This section also discusses how reconfiguring the system at sub-optimal periods can give undesirable performance.

4.2 Static Reconfiguration

As described in Chapter 3 *static reconfiguration* is associated with the class of applications for which the bandwidth requirements are deterministic in nature. There are a total of 12 applications that were simulated of which 2 are real and the other 10 synthetic.

The system was reconfigured at the end of each communication phase, the computation phase was not simulated. It was assumed here that the system could be reconfigured during the computation phase of the application, and hence no delay overhead was incorporated for reconfiguring the interconnect. The set of experiments described in Chapter 3 Section 3.4.1 were simulated for all the communication phases within the 12 applications.

4.2.1 Real Applications

The 2 real applications that were simulated were the SAR (Synthetic Aperture Radar) and the Beam Forming applications. The SAR application that was simulated is described in detail Chapter 3, Section 3.4.1.

The second real application was the beam forming application, which consisted of five communication phases, the first two phases are the same as in the SAR application. The third phase is a partial *reduce* in which node 7 is the destination node, which in the fourth phase *broadcasts* a fraction of its messages to the other 5 processing nodes. The last (fifth) phase is the same as the third phase of the SAR application.

The two real applications simulated have most of the common communication patterns like *broadcast*, *all-to-all* and the *reduce* which are discussed in chapter 3, Section 3.4.1. The one communication pattern that was not present in the two real applications was the *point-to-point* communication pattern.

4.2.2 Synthetic Applications

As described in Chapter 3, Section 3.4.1 the synthetic applications were randomly derived from the set of communication patterns, number of phases, size of the message and the nodes participating in the communication phase.

Each of these ten synthetic applications were simulated the same way as the real applications. The details of the ten applications are summarized in Table 3.1.

4.2.3 Performance Analysis

The performance metrics used for analyzing the results obtained in the various experiments include both the the delay or latency experienced by the flows within the system and the variability in the delay.

The delay in the system is measured in terms of the amount of time a particular communication pattern takes to complete. This is measured from the time the message is created in the system to the time it is delivered to its destination. An alternate form of delay measurement is *speedup*, which for the application or the communication phase is defined as the ratio of the time taken to complete under uniform allocation of resources and the time taken with reconfiguration.

$$Speedup = \frac{Completion\ time\ _{UniformAllocation}}{Completion\ time\ _{reconfigured}}$$

The maximum and mean completion times (across flows within an individual communication phase) are of interest both in absolute terms and as a speedup relative to the uniform bandwidth allocation. The variability measure of this system is obtained as the coefficient of variation of the delay in delivering the messages. The variability is a measure of the fairness of the system, with values near zero implying equal delivery time and values approaching (or exceeding) one indicate variability of the same order as the mean completion time.

$$Coeff\ \ Var = \frac{\sigma\ _{completion\ time}}{\mu\ _{completion\ time}}$$

Results

The only communication pattern that requests variable communications volume across the message set is the point-to-point pattern. The DRR fairness protocol implemented within subrings is effective only for flows which have different message sizes. As a result, to evaluate the performance of the DRR protocol only the point-to-point communication pattern is considered.

Since the DRR protocol reallocates the unused portion of the bandwidth among the active nodes (relative allocation of bandwidth), the maximum completion time of the phase should remain unaffected by its presence, where as the variability in the completion times of the individual flows should decrease. The LCA reconfiguration, on the other hand, impacts the total bandwidth allocated to a particular subring and hence should decrease the maximum completion time of a phase and potentially reduce its variability, too.

There were a total of 14 point-to-point communication patterns (Table 3.1) simulated, the results from which are presented first. Figure 4.1 shows the mean and the maximum completion times for the point-to-point communications phases. The 14 sets of 4 bars each correspond to the 14 different point-to-point communication phases that are present in the 12 applications that were simulated. The 4 bars for each entry correspond to the performance of the system with uniform allocation, using DRR reconfiguration, using LCA reconfiguration , and when using both DRR and LCA for reconfiguration. The *dot* on

the line corresponds to the mean completion time of the phase and the height of the bar is the maximum completion time for the flows in that phase.

As expected and discussed later, presence of the DRR fairness layer increases the mean completion time associated with individual phases and does not affect the maximum completion time. The LCA reconfiguration algorithm significantly decreases the maximum completion time and also the mean completion time. It is also seen that, with both LCA and DRR, the mean and maximum completion times are very close together.

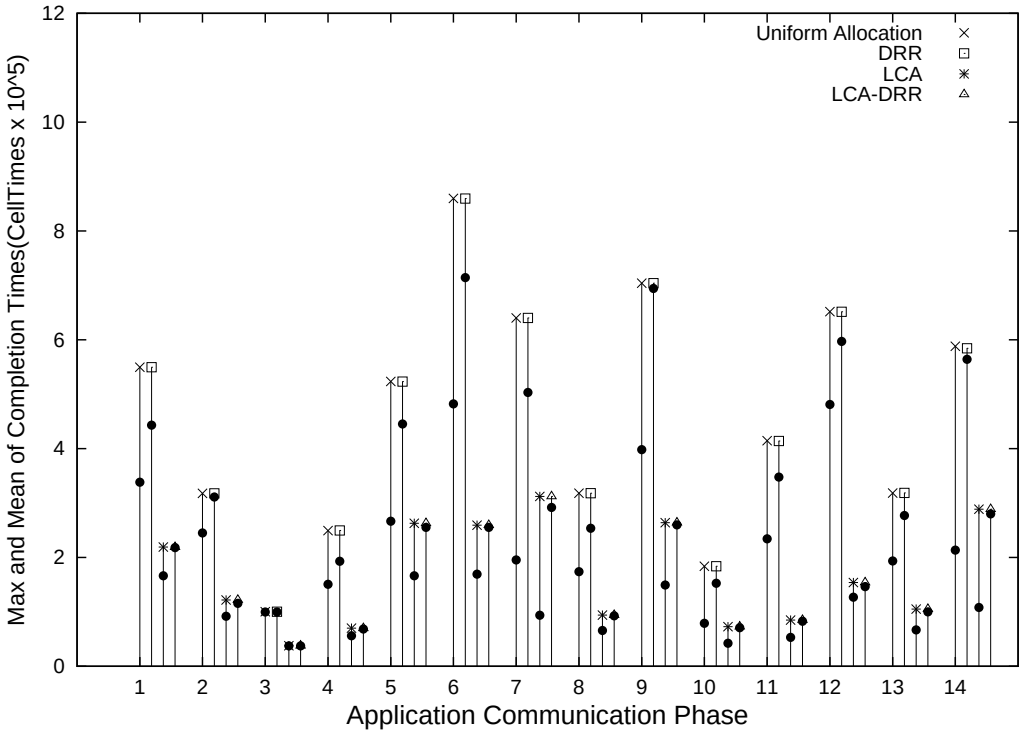

Figure 4.1: Maximum and mean completion time for point-to-point communication phases

The purpose of the DRR reconfiguration is better illustrated in Figure 4.2, which plots the coefficient of variation of the latency in each communication phase. The layout of this figure is the same as the previous one, with respect to the 14 sets. The height of the bars in this figure correspond to the coefficient of variation. There is a significant improvement in the *fairness* of the system with reconfiguration using the DRR protocol. Figure 4.2 also shows that with both the LCA and DRR reconfiguration techniques, the variability in the latency becomes very close to zero, which as discussed earlier characterizes a fair system. The two figures (4.1, 4.2) clearly illustrate the trade-off associated with using DRR. That is, use of DRR assures greater fairness in bandwidth allocation, and thus a reduction in variability, however at the cost of increasing the mean completion time. The

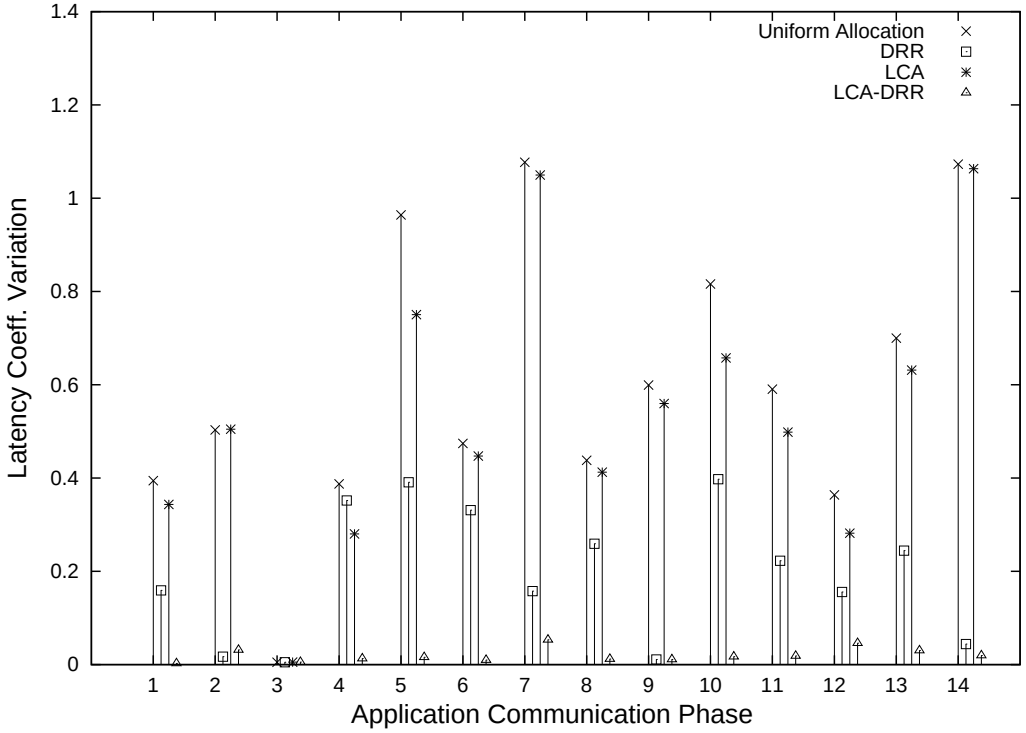

Figure 4.2: Variability in completion times for point-to-point communication phases

LCA reconfiguration on the other hand, has a limited impact on variability, but a dramatic impact on maximum completion time and, in turn, potentially the mean completion time. When the two reconfiguration techniques are combined, there is a combined benefit of a significant decrease in maximum completion time and variability reduced to near zero.

When all of the communication traffic is considered, the performance metric used is the *speedup*. All the communication phases within a application are taken into account and a speedup number for the communications required by the application is obtained. Figure 4.3 shows the communication completion time for all the applications simulated. The comparison is made between the Uniform Allocation and the LCA-DRR (both LCA and DRR) reconfigurations. In each of the applications there is a significant decrease in time associated with the communication phases. The speedup obtained in the communication phases of the applications simulated varies from 1.9 to 7.1 (Table 4.1) . The average speedup obtained across all the applications is approximately 4. The large variation in the speedup across the applications is because not all applications have the same communication patterns, and the speedup is dependent on the type of communication pattern and the amount of traffic associated with the particular pattern. Figure 4.4 shows the maximum, median and minimum speedup that each communication pattern yields across the entire set of applications. Understandably the reduce communication pattern derives the most

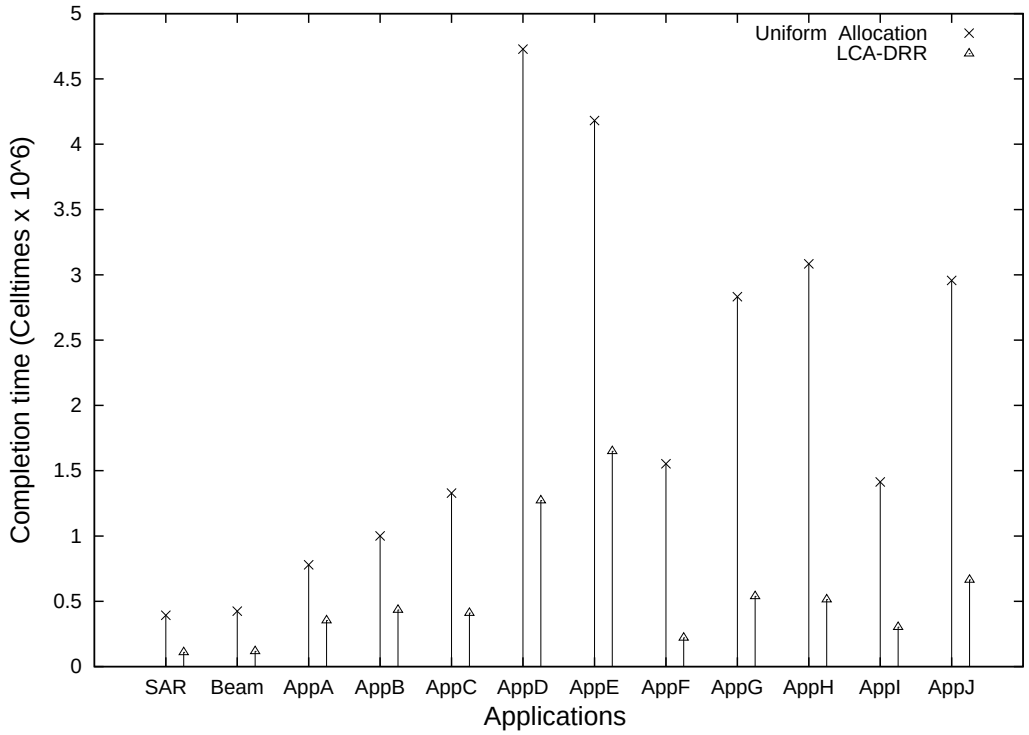

Figure 4.3: Communication phase completion times across applications (with and without reconfiguration)

benefit with respect to speedup, as all of the communication bandwidth is allocated to a single subring (i.e., the destination node of the reduce phase). There is a significant improvement seen in all the other communication patterns too, which also supports the idea of reconfiguration in such applications.

The performance numbers presented so far have exclusively represented the communication phase of an application. For the overall speedup of an application, a model which incorporates both the communication as well as the computation numbers is discussed next.

In the applications simulated, and in signal processing applications in general, the computation and communication phases are often mutually exclusive, so for an overall speedup of the application we can make use of Amdahl's Law. Here,

$$Speedup_{overall} = \frac{1}{f_{comp} + \frac{f_{comm}}{Speedup_{comm}}}$$

where f_{comp} is the fraction of original execution time associated with the computation and f_{comm} is the fraction of the original execution time associated with communications. The factor $Speedup_{comm}$ is the communications speedup obtained from the simulated applications. Knowing the ratio of communication time to computation time the same can be

Table 4.1: Completion time data for the 12 applications simulated

Application	Uniform Allocation (celltimes)	DRR-LCA Reconfiguration (celltimes)	Speedup
(SAR)	393000	109000	3.60
(Beam)	424000	117000	3.62
AppA	779000	353000	2.20
AppB	1001000	434000	2.30
AppC	1329000	410000	3.24
AppD	4728000	1271000	3.72
AppE	4181000	1649000	2.53
AppF	1553000	220000	7.05
AppG	2832000	538000	5.26
AppH	3084000	513000	6.01
AppI	1414000	302000	4.68
AppJ	2957000	664000	4.45

expressed as

$$Speedup_{overall} = \frac{1+R}{1+\frac{R}{Speedup_{comm}}}$$

where, $R = \frac{Tcomm}{Tcomp}$.

Figure 4.5 illustrates the overall speedup in applications simulated with the model described earlier. The communication to computation ratio is plotted from 0.1 to 10 to span two orders of magnitude. The three curves represent the minimum, mean and maximum speedup in communications completion time across the applications. From the figure it is evident that only enhancing a part of the overall execution time of the application gives limited improvement when the entire picture is considered. Though this is true, it should also be noted that even in the interval of 0.5 to 2 for $\frac{T_{comm}}{T_{comp}}$, which is typical for the signal processing applications, there is a 20% performance gain predicted under fairly pessimistic assumptions and on the other hand close to 100% gain is potentially attainable.

4.3 Dynamic Reconfiguration Results

As described in chapter 3 dynamic reconfiguration is associated with reconfiguring the interconnect in a network switching system. The mechanism for generating self-similar traffic is adapted from [37]. It involves transfer of files, the sizes of which are drawn from an *heavy-tailed* distribution, characterized by

$$P[X > x] \sim x^{-\alpha} \quad x \to \infty$$

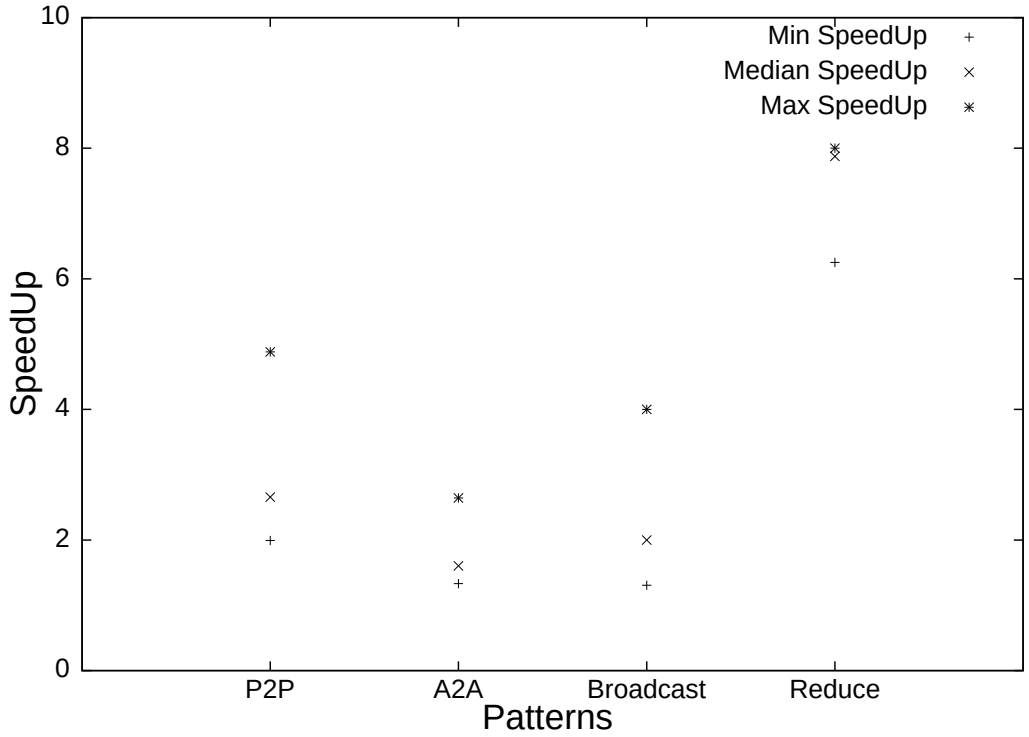

Figure 4.4: Maximum, median and minimum speedup obtained across communication patterns.

where $0 < \alpha < 2$. The heavy-tailed distributions were obtained from the *Pareto* distribution, with a probability density function given by

$$p(x) = \alpha k^{\alpha} x^{-\alpha-1}$$

where $\alpha, k > 0$ and $x \geq k$. The distribution function has the form

$$F(x) = P[X \leq x] = 1 - (k/x)^{\alpha}$$

where the parameter k represents the smallest possible value of the random variable.

For the experimental cases simulated, a value of $\alpha = 1.05$ was chosen, which corresponds to a very high degree of *self-similarity* [37]. The mean file size or the mean of the distribution was chosen to be $\approx 4.1KB$[1]. The load on the interconnect is varied by varying the inter-arrival times of the burst (i.e. "OFF" period).

To enable proper operation of the access protocol, bursts originating from the nodes are divided into packets or messages which are a maximum of 1000 cells. The DRR gives access to the link based on the *packetsize* of the packet at the beginning of each queue in

[1]This particular file size is chosen based on ideas in [37].

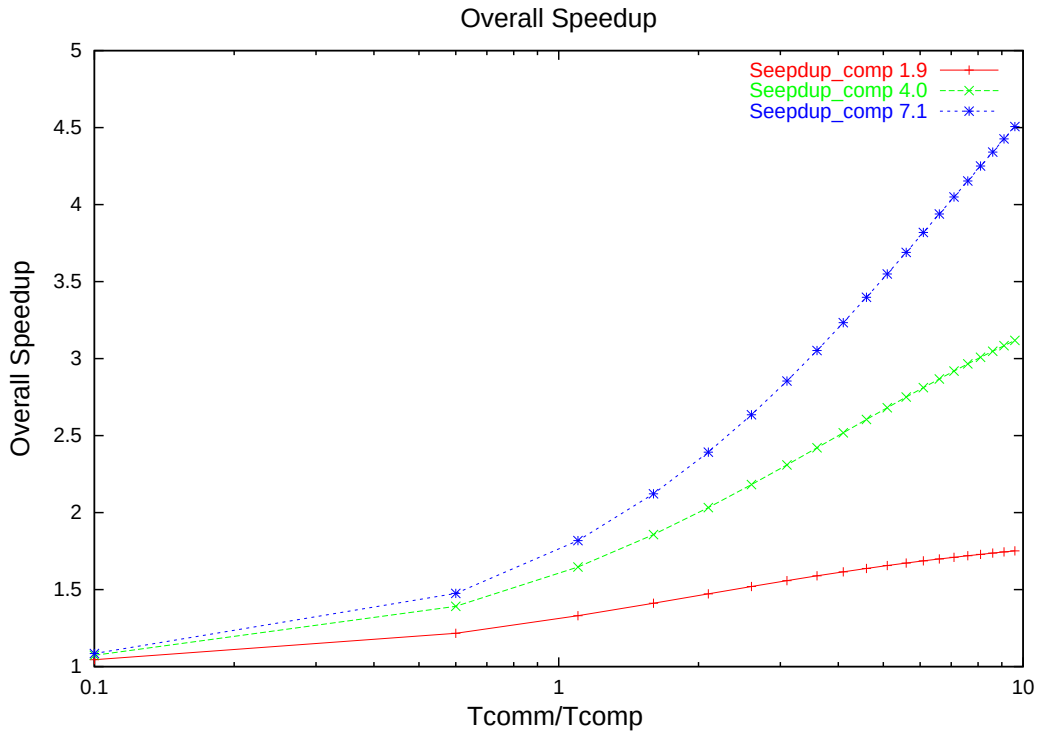

Figure 4.5: Overall performance improvement.

the link. This ensures that all the flows have the opportunity to share the link, even when a particular flow has a huge burst to send to a particular destination.

4.3.1 Performance Analysis

In the case of *Dynamic Reconfiguration* we allocate bandwidth to flows based on perceived demand at each of the reconfiguration times. We model a memoryless control system in which the bandwidth requirement for each flow is characterized by its input queue lengths. Bandwidth allocations are proportional to the queue lengths, which effectively encourages the system to equalize completion times across all the sources within a subring and also across the subrings.

The Laser Channel Allocation(LCA) ensures that all the subrings get bandwidth proportional to the queue lengths for the packet destined to their terminal nodes, and the DRR ensures fairness among the sources within the subring. As described the system is reconfigured for equal completion times, thus we expect a decrease in the variability in packet delivery times.

We will illustrate the operation of the system with trace data from an individual simulation experiment (i.e., set of source messages). This input was simulated with no reconfiguration periods. Figure 4.6 illustrates the instantaneous queue lengths at each of

the output ports both with no reconfiguration and with reconfiguration at every million, hundred thousand, and thousand celltimes periods, in that order. These correspond to simulations of $\rho = 0.5$ (50% utilization) in the system.

Figure 4.6a corresponds to the case when there is no reconfiguration (i.e., uniform allocation) during the entire run. We see that the dequeue rate (observed as the slope of the lines) is almost the same for all the subrings. It also shows that there is a large disparity in the number of cells queued for a subring with a large burst and those which did not have a huge burst. The effect of the burst can be seen dominating the characteristic for the duration of the simulation, which is not desirable. This is in contrast to Figure 4.6b where the subrings with higher load in the queue get higher bandwidth and hence are characterized by steeper slopes. In this figure there is a considerable decrease in the amount of time that the burst in a particular subring is actually present. Also the number of cells queued at the end of the simulation (10^7 cell times) is significantly less compared to the uniform allocation case, and each of the sources also have approximately the same number of cells in their queues. We see from Figures 4.6c and 4.6d that as we increase the reconfiguration rate (decrease the reconfiguration period), the detection of the burst and also the time when the effect of the burst diminishes is identified faster, i.e., the response of the system is faster and, as we will see, this benefits the delay in the system.

From the plots, it is clear that the average queue length over all the sources is significantly lower for the reconfigured cases. This is quantified in Table 4.2 where the average queue lengths over the entire simulation are given for each of the destination nodes. These results are from a set of runs, rather than an individual input. Over all the nodes there is a reduction of 63.81% reduction in queue length for the hundred thousand celltime reconfigured case relative to the uniform allocation case.

Table 4.2: Mean queue lengths (cells)

	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6	Port 7	Port 8	Avg.
Uniform	400353	337703	259190	89206	335597	526592	486955	535545	371268
100,000	127756	164486	93259	89843	109855	142086	243290	104304	134358
%Improv.	68.09	51.30	64.02	-0.71	67.27	73.02	50.04	80.49	63.81

A thousand celltimes was the smallest reconfiguration period simulated because the underlying signaling mechanism for controlling the data and control cells in the system has a reset period of a thousand cells [28]. Furthermore a burst of cells are divided into packets with a maximum size of a thousand cells, and as system reconfiguration does not affect the packet that is in transit from a source to destination port, a thousand cell time period is a reasonable approximation to an optimal reconfiguration period.

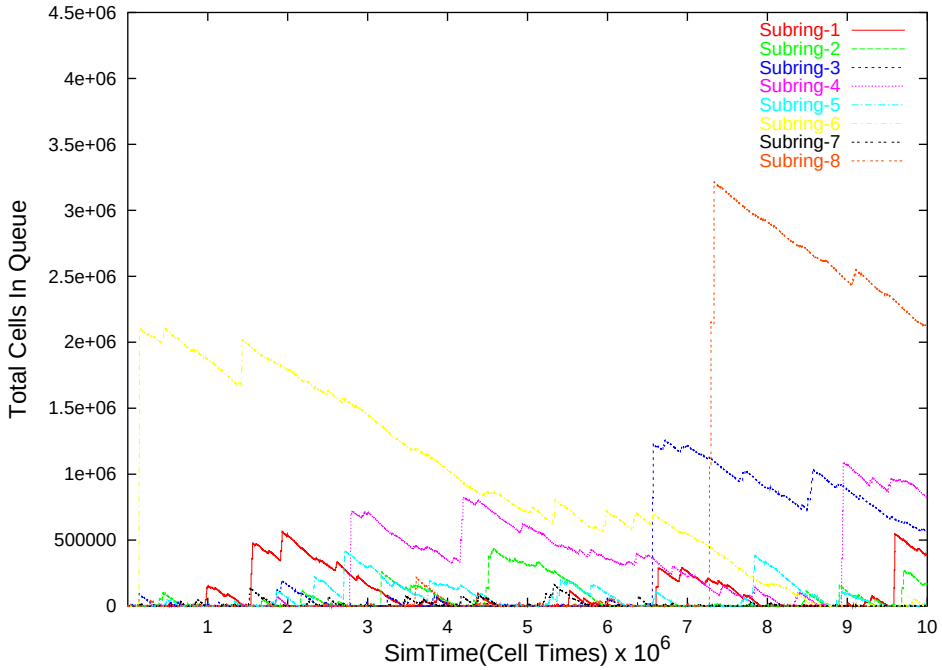

Figure 4.6a: Uniform Allocation

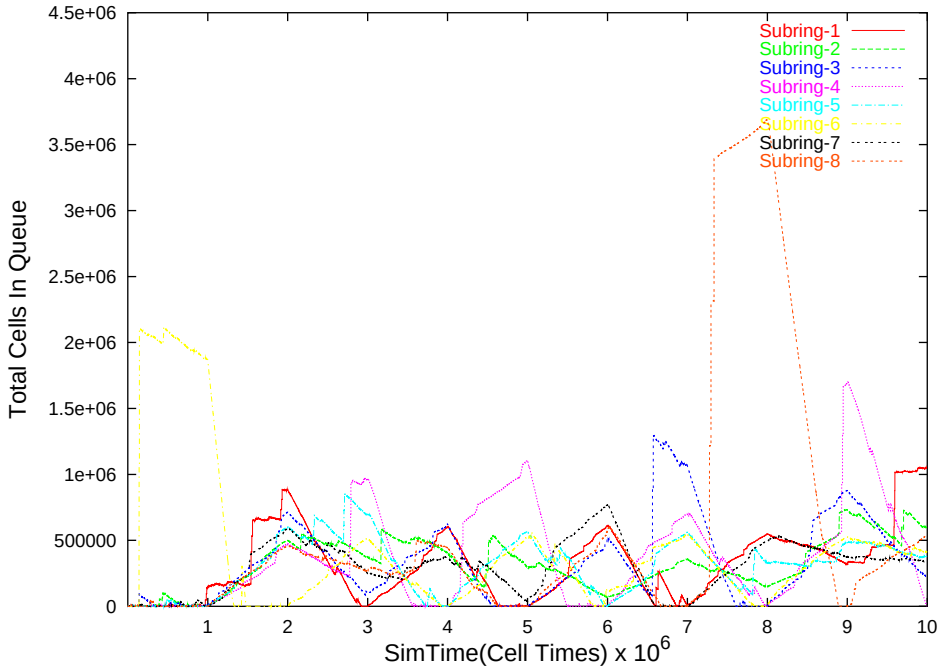

Figure 4.6b: Reconfigured - Million celltime period

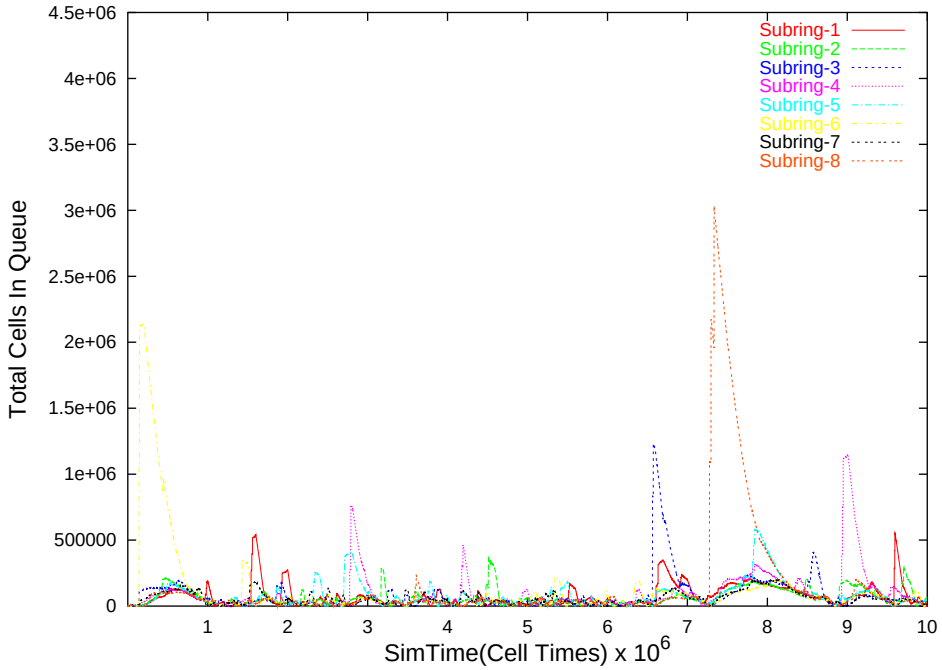

Figure 4.6c: Reconfigured - Hundred thousand celltime period

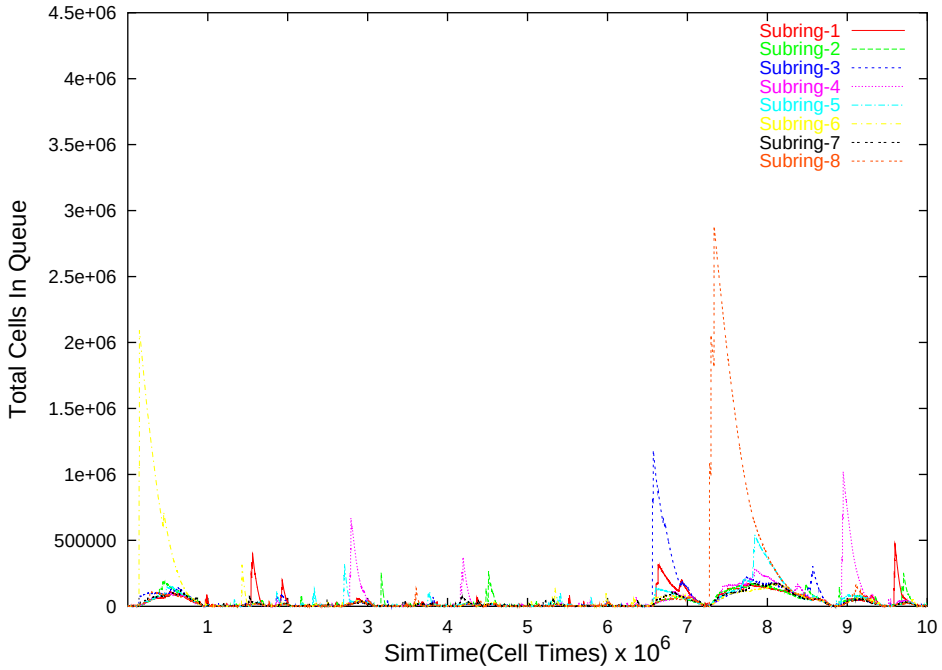

Figure 4.6d: Reconfigured - Thousand celltime period

Figure 4.7 illustrates the average packet delay over the entire system and also the average delay across individual subrings. Figure 4.7a shows the overall packet delay, accumulated from 10 different runs. The overall packet delay for the million celltime reconfiguration case is higher than the uniform allocation case. The hundred thousand and thousand celltime reconfiguration cases, on the other hand, give an improvement in the mean packet delay in the system.

Figure 4.7b gives the delay numbers for the individual subrings across 10 runs. We see here that the spread of these numbers in the uniform allocation is much higher than the reconfigured cases, which is expected, because of the control algorithm controlling the variability in the reconfigured runs.

The degradation in mean packet delay performance for the million celltime reconfiguration is a result that is common to memoryless control systems. The system is configured based on demand at a particular point in time. At a later time, however, the demand is potentially significantly altered, and unless the control system reacts to the demand change, the system itself is poorly configured to service the actual demand present. An illustration of this effect is shown in Figure 4.6b, which shows the instantaneous queue lengths for a million celltime reconfiguration period.

We see that at time 7×10^6 when the system is reconfigured, the number of packets in queue for port 8 is quite small. This will imply the bandwidth allocated to that port is low. This port then receives a huge burst after a short time, which we see is not detected by the control algorithm until the next period (at time 8×10^6). Until this later reconfiguration, few packets in this queue are serviced, which is seen by the increasing slope of the queue length for destination 8. This increases the average packet delivery time due to a poorly configured system. Further analytical analysis of this effect is presented in Section 4.3.2.

One metric that is of interest is the *Speedup* of the reconfigurable system over the *uniformly allocated system*. The speedup is defined as:

$$S = \frac{AverageDelay_{Uniform}}{AverageDelay_{Reconfig}}$$

For the hundred thousand and the thousand celltime reconfiguration cases the speedups are 1.71 and 2.22, respectively, when comparing the average packet delay for the overall system. The million celltime reconfiguration case has a speedup of 0.57, suggesting that a million celltime reconfiguration period is clearly inappropriate for this system.

Another important performance metric is the variability in packet delay. Given the control algorithm in use, we expect a significant reduction in the delay variability for the reconfigured system. Figure 4.8a shows the variability in packet delivery across all the subrings. It shows the standard deviation of packet delay for the overall switch. The minimum, mean and maximum standard deviation from the different runs are plotted.

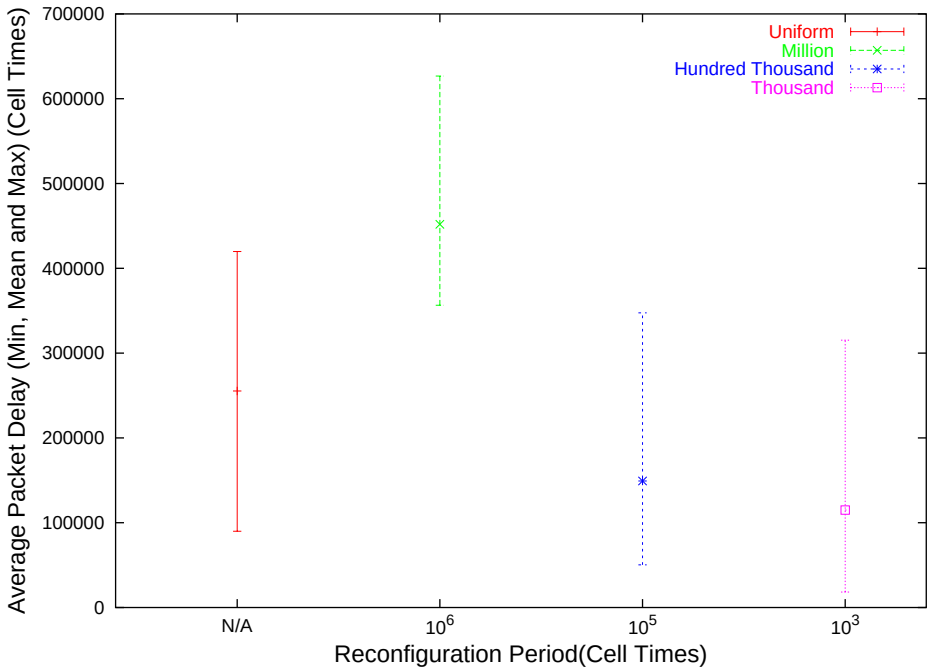

Figure 4.7a: Average delay in packet delivery in the system (overall)

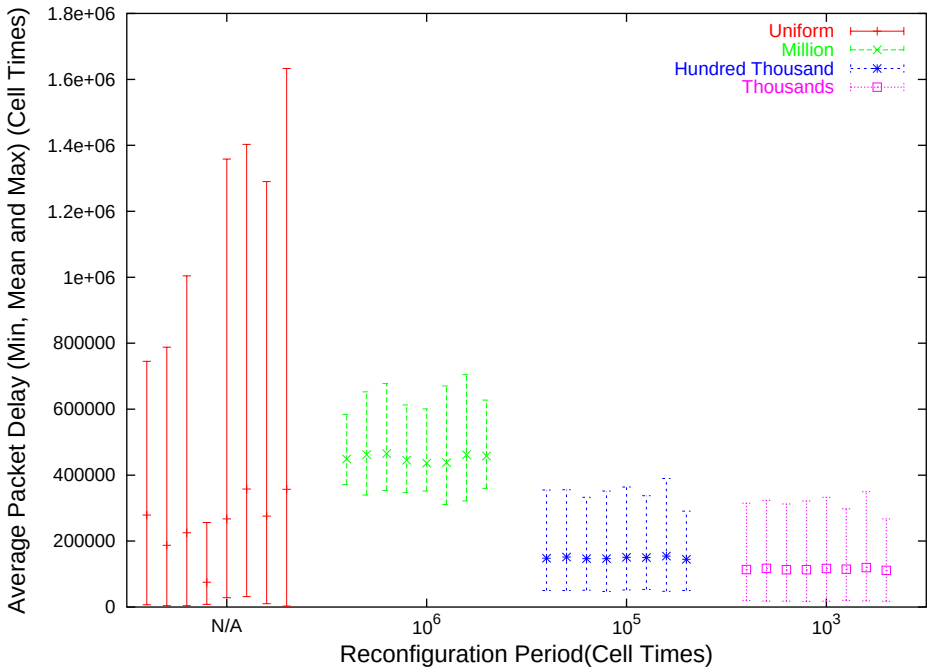

Figure 4.7b: Average delay in packet delivery in the system (subring)

The figure presents the comparison between the uniformly allocated system and when the system is reconfigured with three different reconfiguration periods. The plots shown here correspond to $\rho = 0.5$ (i.e., 50% load in the system) and are aggregated over 10 runs.

Our reconfiguration mechanism as mentioned earlier makes the bandwidth allocations proportional to the load on individual flows. Thus if the packets at all the sources are drained after any of the reconfiguration time slot, they would finish at the same time. The delay experienced by these packets before they are delivered is about the same for all the flows. Expectedly, the variability in packets delivery decreases with the number of times the system is reconfigured. Figure 4.8a illustrates the same point, the million reconfiguration period has a lower variability compared to the on reconfiguration case. The trend continues for the hundred thousand and thousand reconfiguration periods, though the difference is not as significant as the difference between the million and the hundred thousand case.

Table 4.3 shows the standard deviation of packet delay over the entire set of 10 runs. The improvement in variability of delay is also apparent here.

Table 4.3: Overall standard deviation over all runs

	Base Case (celltime)	Million Celltime (celltime)	Hundred Thousand Celltime (celltime)	Thousand Celltime (celltime)
Std. Dev.	2.685×10^6	0.33×10^6	0.2293×10^6	0.2285×10^6

The patterns of decreasing mean standard deviation and spread are also consistent when individual subrings are considered. Figure 4.8b illustrates these numbers for individual subrings. The 4 sets in the plots correspond to uniform allocation, million celltime, hundred thousand celltime and thousand celltime reconfiguration periods. The 8 bars within each set correspond to the 8 subrings in the multiring architecture. We also see that the variability between the subrings themselves is considerably decreased.

As in the static reconfiguration case, the DRR algorithm controls the variability within a subring. This is demonstrated by the Figure 4.9, which compares the uniform allocation case to the case when only DRR reallocation is made. The plots consists of 8 sets of 7 points, where each set corresponds to one of the output ports and the 7 points correspond to the 7 sources delivering packets to each output port. It is easily seen that the delays within a subring have a lower variability. However the DRR allocation does not affect the bandwidth available within a subring, and thus doesn't impact packet delay across subrings.

LCA, on the other hand, does not affect the variability within a subring. LCA in this control algorithm tries to make the utilization across the system uniform. This effect is shown Figure 4.10, which compares the average packet delay for each subring in the uniform allocation case to the case when only LCA reconfiguration is employed. For both the DRR only and LCA reconfiguration, the system was reconfigured every 100,000 celltimes.

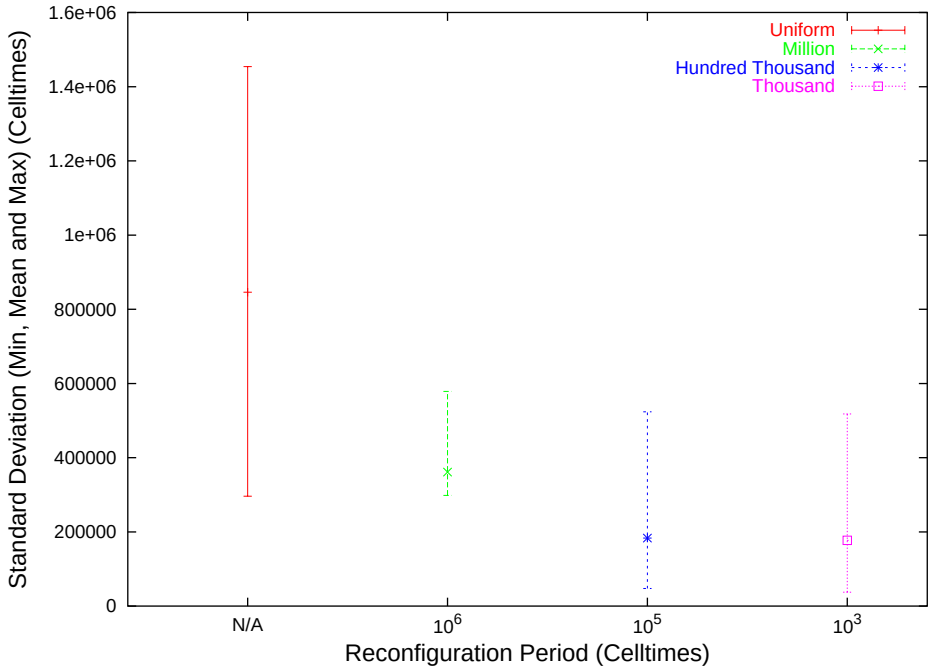

Figure 4.8a: Standard deviation of packet delay across all subrings

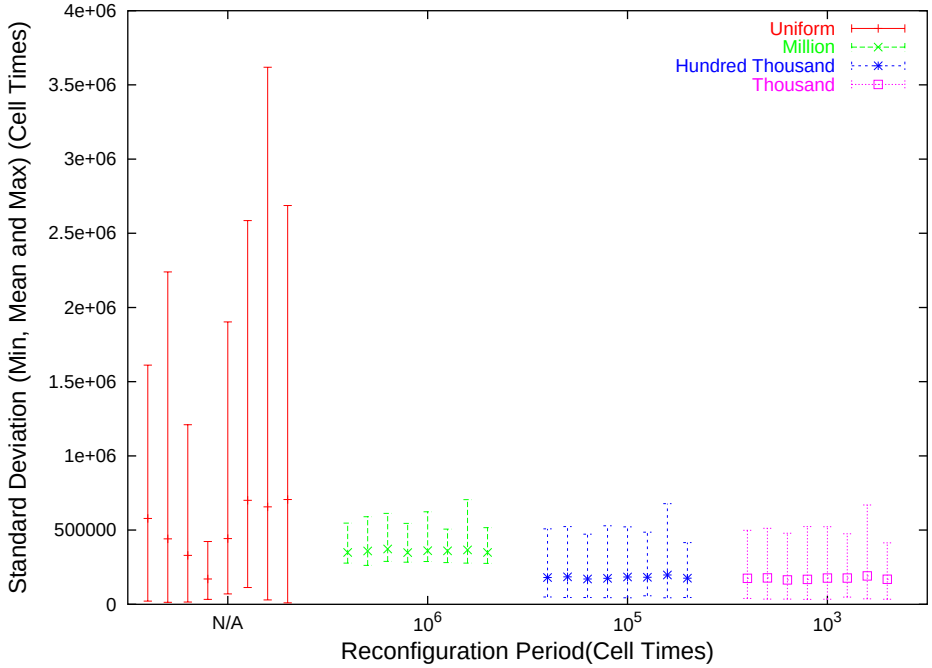

Figure 4.8b: Standard deviation of packet delay in individual subrings

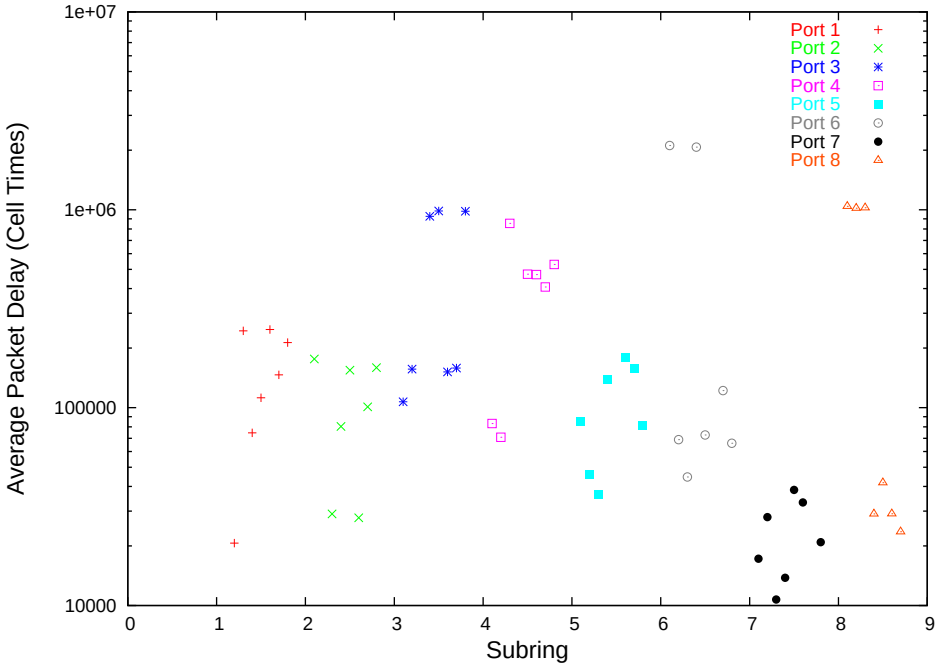

Figure 4.9a: Average Packet Delay for each source in each subring (Uniform)

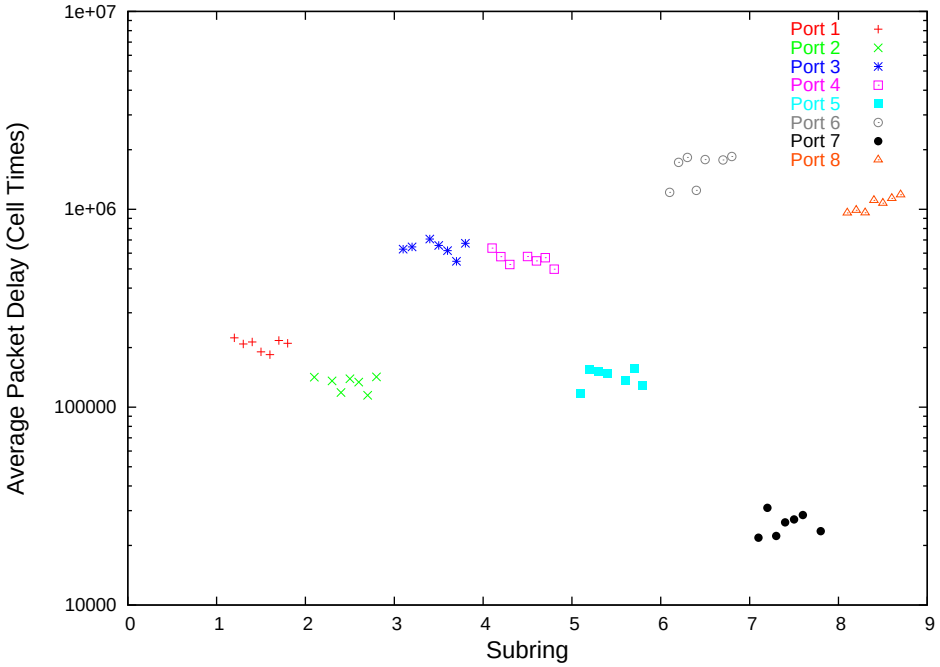

Figure 4.9b: Average Packet Delay for each source in each subring (DRR only)

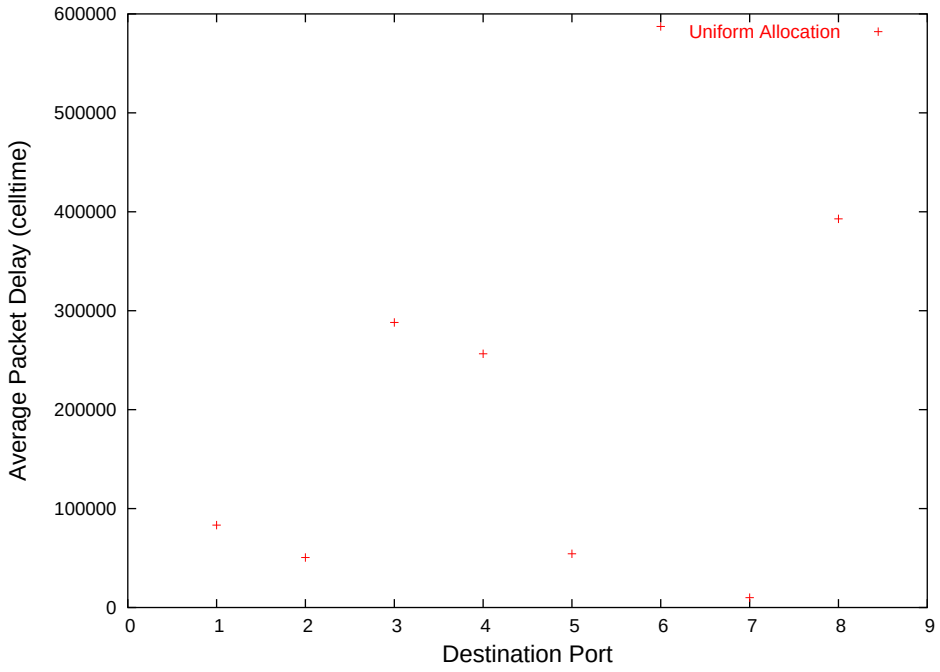

Figure 4.10a: Average Message Delay on each subring (Uniform)

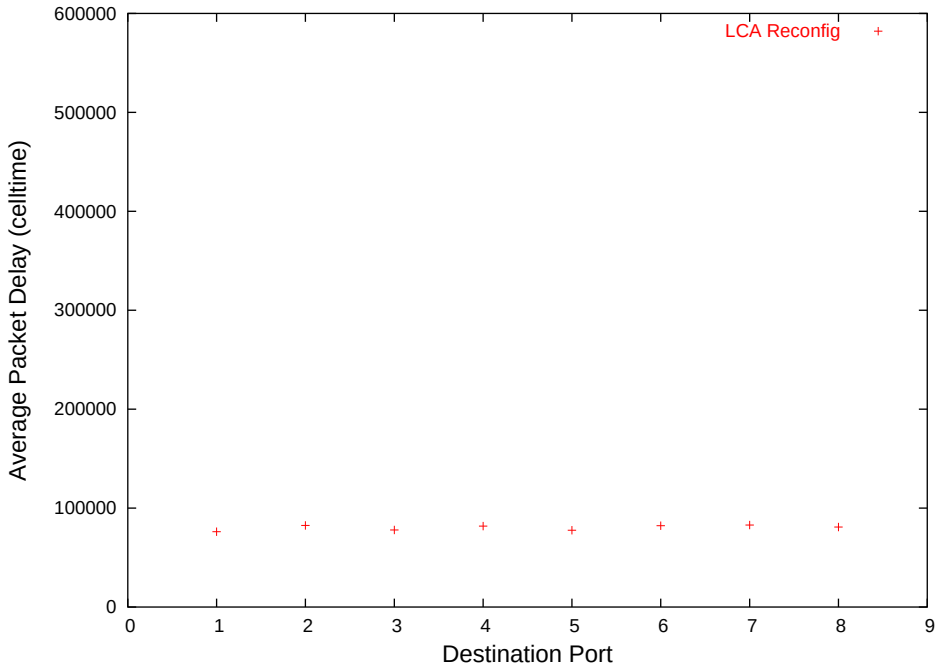

Figure 4.10b: Average Message Delay on each subring (LCA only)

4.3.2 Reconfiguration at sub-optimal period

As seen in Figure 4.7a, the case where the system is reconfigured every million celltimes performs poorly when compared to the uniform allocation. Though this seems counterintuitive, it can be shown analytically that a large reconfiguration period can in fact increase the mean delay through the system. Let us consider a M/D/1 system, which approximately represents each subring of the multiring architecture in consideration. Let us assume that exactly one of the 8 subrings gets a burst of 2×10^6 cells and the other 7 subrings have a Poisson arrival with $\lambda = 1$ and a deterministic service rate of $\mu = 2$. Also the generators stop generating traffic 10^6 celltimes after the burst.

Uniform Allocation

For a M/D/1 queue the average wait time in the system is

$$D_{M/D/1} = \frac{1}{2}\frac{\rho\mu^{-1}}{1-\rho}$$

thus for the seven subrings which do obey the above equation the delay is

$$D_{M/D/1} = \frac{1}{4}$$

The delay for the bursty subring will be dominated by the burst. Thus the delay for the bursty subring can be approximated as the average delay experienced by the cells in the burst. We can assume that the packets will be delivered at $0.5, 1, 1.5, ..., 1 \times 10^6$ celltimes. The total delay in the system is then

$$TotalDelay(D) = 0.5 \times \Sigma_{i=1}^{2\times10^6} i$$

The average delay of the cells in the burst is then obtained as the ratio of D to the total number of cells in the burst which is,

$$D_{burst} = \frac{0.5 \times \Sigma_{i=1}^{2\times10^6} i}{2 \times 10^6} = \frac{1}{2} \times 10^6$$

Thus the weighted delay of the system as a whole is given by

$$D_{overall} = \frac{((7 \times D_{m/D/1}) \times (1 \times 10^6)) + ((1 \times D_{burst}) \times (2 \times 10^6))}{(7 \times 1 + 1 \times 2) \times 10^6} \approx \frac{1}{36} \times 10^6$$

Reconfigured every million cell times

Let us now assume that the service rate for the bursty subring is so reconfigured that the service rate for the other seven subrings becomes a third of the original i.e., $\mu = \frac{2}{3}$. This makes these subrings have a utilization of more than unity, which makes these overloaded too. If we assume that the cells are generated by unit time i.e., $0, 1, 2, ...$ then they will be delivered at $1.5, 2, ...$ The average delay for these subrings now becomes,

$$D_{overloaded} = \frac{1.5 + 2 + ... + (1.5 + 0.5 \times (10^6 - 1))}{1 \times 10^6} \approx \frac{1}{4} \times 10^6$$

The average delay for the bursty subring with the new service rate $\mu_{reconfig} = \frac{34}{3}$ becomes

$$D_{burst(reconfig)} = \frac{\frac{3}{34} \times \Sigma_{i=1}^{2 \times 10^6} i}{2 \times 10^6} = \frac{3}{44} \times 10^6$$

The overall delay in the reconfigured case now becomes,

$$D_{overall(reconfig)} = \frac{((7 \times D_{overloaded}) \times 1 \times 10^6) + (1 \times D_{burst(reconfig)}) \times (2 \times 10^6)}{9 \times 10^6} \approx \frac{7}{36} \times 10^6$$

which is 7 times worse than the uniform allocation case.

This clearly demonstrates that a poorly chosen reconfiguration period can increase the mean delay in the system compared to the uniform allocation case. Although the above model is not a strict representation of the system under consideration, it gives us an idea as to why there is the degradation in the performance of the million celltime reconfiguration case.

The dynamic reconfiguration performance results shown to this point don't consider the cost of reconfiguring the system. In this section we present an analytic model that includes this overhead in a performance prediction of mean delay.

In this analytical model the cost of reconfiguration is added after the simulation. Using *Little's law* and the mean packet delivery time we obtain an estimate of the number of packets that are present in the system during each reconfiguration. The average reconfiguration penalty per packet (R_c) can be derived as

$$R_c = \frac{Q_t \times P \times M}{N}$$

where Q_t is the mean queue length during the simulation run, P is the reconfiguration penalty, M is the number of times the system is reconfigured during the entire simulation run and N is the total number of packets delivered during the entire simulation run. From Little's law we know that $Q_t = \lambda \times W_t$, where W_t is the wait time in the system and λ is the mean arrival rate. Using the numbers we have from the simulation, the expression for

R_c now becomes

$$R_c = \frac{\lambda \times W_t \times P \times M}{N}$$

where $\lambda = \frac{N}{Simulationduration}$. This now reduces the expression for R_c to

$$R_c = \frac{W_t \times P \times M}{10^7}$$

R_c represents the cost of reconfiguring the system, i.e., the delay that is to be added to the average packet delay in the system shown earlier, such that it includes the time taken to change the bandwidth allocation in the multiring.

Relevancy of Little's Law

We have earlier argued that steady state assumptions do not hold for these simulations. However, Little's law is valid only under steady state conditions implying that using it to obtain the mean queue length is not strictly appropriate. To estimate the error introduced by using Little's relationship we ran a few simulations in which a delay histogram with a mean bin size of 1000 celltimes was constructed. In these bins we assumed that the mean packet delay was the mean of the bin itself. We calculated the probability of a packet with this mean packet delay for the bin being in the system during the reconfigurations. With this number, the number of packets in each bin, and the penalty P associated with each reconfiguration, we calculated the reconfiguration penalty.

Table 4.4: Reconfiguration penalty per packet (R_c) (comparison between simulation model and Little's Law)

RP (celltime)	10^6	10^6	10^5	10^5	10^3	10^3
	(histogram)	(Little's)	(histogram)	(Little's)	(histogram)	(Little's)
run 1 (celltimes)	0.42580	0.42575	0.87331	0.87295	53.40197	53.37588
run 2 (celltimes)	0.58895	0.58890	2.86251	2.86218	248.02417	248.00300

The results are tabulated in Table 4.4, which shows the average reconfiguration cost per packet for both Little's law and the histogram that results from the simulation for the four different reconfiguration periods (RPs). When comparing the numbers in Table 4.4 we can conclude that Little's law gives us a reasonable approximation. The numbers presented in the table correspond to a reconfiguration penalty of 1 celltime, i.e., it takes 1 celltime to reconfigure the interconnect. Figure 4.11 shows the mean delay in a system under different reconfiguration penalties.

Figure 4.11 shows the effect of adding reconfiguration cost to the mean delay numbers. This figure corresponds to reconfiguration times in the range $0 - 1000$ celltimes. This corresponds to a range of $(0 - 4\mu s)$, which is a good estimate for the time to reconfigure

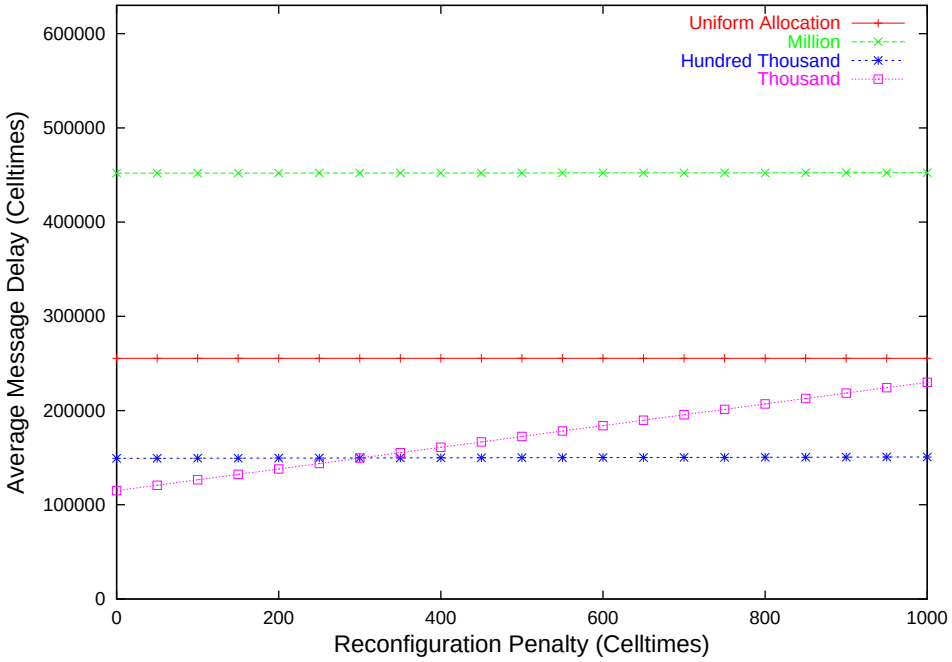

Figure 4.11: Average packet delay across all subrings

the system, as in the worst case we will have to drain the packets in the system to synchronize the configuration at all the nodes. It is seen that a system with million celltime and hundred thousand celltime reconfiguration periods are not affected much by the reconfiguration penalty i.e., they have almost flat lines for the entire range. This does not hold for the thousand celltime reconfiguration, as there are a large numbers of reconfigurations of the system. The benefits obtained by reconfiguring the system can easily be consumed by the overhead cost. As shown in Figure 4.11 the thousand celltime reconfiguration case gives the same performance as the hundred thousand case at around 300 celltime penalty for reconfiguration and is even worse at higher penalties. It tends to approach the uniform allocation case at around 1200 celltime penalty.

The above performance results tend to point toward a reconfiguration period of about $100,000$ celltimes. It does reduce the variability in the packet delivery, which was the purpose of having a reconfigured system, and also in the process decreases the delay per packet. It is preferred over the million celltime reconfiguration period due to the latter's increase in average packet delivery delay and the thousand celltime reconfiguration performs poorly when we account for the reconfiguration penalties.

4.4 Summary

The performance implications of reconfiguring an optical interconnect are obtained for both static and dynamic reconfigurations.

Statically reconfiguring the interconnect for the signal processing class of applications we obtained a speedup of 1.9 to 7.1 for the communication phases. This corresponds to overall performance gains from 20% to 100% for the application set.

Dynamic reconfiguration was employed to improve the performance of the interconnect in a switch fabric. The speedup of the system, which is measured as the ratio of the average packet delay under uniform allocation to the reconfigured case, ranges from 0.57 to 2.22. A speedup less than unity for the million celltime reconfiguration case demonstrates that a poorly chosen reconfiguration period can have undesirable effects on system characteristics. A dramatic improvement in delay standard deviation was also seen for all reconfiguration periods. A hundred thousand celltime was inferred as an appropriate reconfiguration period, as it is possible to obtain a reasonable speedup without burdening the system with frequent reconfigurations.

Chapter 5

Conclusion and Future work

This chapter presents an overview of the contributions of the thesis. This thesis is an investigation into the performance implications of reconfiguring an optical interconnection network. It is an attempt at establishing the benefits of reconfiguring an interconnect according to the demand on it.

Two different set of applications were simulated, one corresponding to the case in which the load on the interconnect was known at compile time, and another in which the requirements can change at execution time. The first corresponds to signal processing applications, where as the latter corresponds to the requirement in a network switch.

5.1 Contributions

Work was initially done to expand the capabilities of the ICNS framework. As of now the ICNS can be used to simulate a true multiring and support reconfiguration. Reconfiguration can be simulated on a per-flow basis. Techniques to incorporate reconfiguration such as Laser Channel Allocation (LCA) and Deficit Round Allocation (DRR) are modeled.

5.1.1 Static Reconfiguration

We observed that many signal processing applications that are run on parallel machines are characterized by alternating communication and communication phases. We went on to model the characteristics of the communication phases in such applications. These were modeled via simulation and the results are presented next.

Static Reconfiguration Performance

Speedups of 1.9 to 7.1 were reported for various signal processing communications phases of the applications. An analytical model based on Amdahl's law was used to established the speedup of the application as a whole. This showed that a speedup of 1.9 to 7.1 corresponds to an overall performance gains ranging from 20% to over 130%.

5.1.2 Dynamic Reconfiguration

Dynamic reconfiguration was used in the context of network switching, where the switch fabric was reconfigured periodically based on the instantaneous demand at each port.

A self-similar input traffic model was used to generate traffic for the simulations. A memoryless dynamic control algorithm, which reconfigures the switch fabric, was modeled in simulation. The simulation results are presented next.

Dynamic Reconfiguration Performance

The simulations for evaluating the performance of Dynamic Reconfiguration also explore the question of setting an appropriate reconfiguration period for the system, in addition to quantifying the performance.

The speedup of the system, which is measured as the ratio of the average packet delay under uniform allocation to the reconfigured case, ranges from 0.57 to 2.22. A speedup less than unity for a million celltime reconfiguration case demonstrates that a poorly chosen reconfiguration period can have undesirable effects on system characteristics. The speedup of 1.71 for the 100,000 celltime reconfiguration period illustrates the clear potential for overall improved performance due to reconfiguration without the need to unduly burden the system with frequent reconfiguration operations. The mean queue length improvement of 63.81% and dramatic improvement in delay standard deviation for this case is further evidence of the appropriateness of this reconfiguration period.

5.2 Summary

This work has investigated two reconfigurations mechanisms for an optical multiring interconnect. Within a ring, the DRR fairness protocol allocates instantaneous bandwidth across the sources contending for an individual destination. If some sources do not utilize their allocated bandwidth, the excess bandwidth is then distributed across the contending sources. Across the multiring, the LCA mechanism supports the flexible assignment of

bandwidth to each ring (and its associated destination). This work shows that the reconfiguring mechanism LCA and DRR can be used to efficiently use the bandwidth available in a system.

In case of parallel applications which are limited by bandwidth limitations, reconfiguring the system has a significant improvement with negligible overhead.

The use of optical chip-to-chip communication enables the construction of a network router switch fabric that can support aggregate throughputs of 1 Tb/s. The ability to reconfigure the fabric enables one to utilize the bandwidth resources even more effectively.

5.3 Future Work

This section describes two possible directions further research can be headed using the available infrastructure. The first is a direct extension which relates to the reconfigurable switch fabric, and the second explores the possibility of using this ICNS based simulator in conjunction with other simulators to model a more general system.

Steady state load characteristics can be derived for the flows in a network switch by aggregating statistics over a considerable period of time. These can then be used to determine the initial configuration of the switch for which the bandwidth allocated for each flow is proportional to the steady state load on the flow, i.e., we configure the switch only once. It would be interesting to evaluate the performance of this model when compared with the uniform allocation and the dynamic reconfiguration policies of bandwidth allocation.

The second direction is related to the idea of *Federated Modeling*. There has been a significant level of interest lately in federated simulation in hierarchical systems. The questions that can be investigated is, how can one integrate this interconnection simulator with other simulators, which model a totally different subsystem, to build an integrated simulation model of the large scale system? It would be interesting to evaluate the human effort in combining these different models and compare it to the scenario when such a simulator is to be built from scratch. Also of interest will be the overall fidelity of such a model, which should address the issue of how accurate a representation of the overall system, does the federated model provide.

References

[1] Alok Aggarwal, Amotz Bar-Noy, Don Coppersmith, Rajiv Ramaswami, Baruch Schieber, and Madhu Sudan. Efficient routing in optical networks. *Journal of the ACM*, 43(6):973–1001, 1996.

[2] D. Bertsekas and J. Tsitsiklis. Parallel and distributed computation, 1989.

[3] Dimitri Bertsekas and Robert Gallager. *Data Networks*. Prentice Hall, second edition, 1992.

[4] G. C. Boisset, M. H. Ayliffe, B. Robertson, R. Lyer, Y. S. Liu, D. V. Plant, D. J. Goodwill, D. Kabal, and D. Pavlasek. Optomechanics for a four-stage hybrid-self-electro-optic-device-based free-space optical backplane. *Applied Optics*, 36:7341–7358, 1997.

[5] A. W. Bojanczyk and J. Lebak. Design and performance of a portable parallel library for stap. *IEEE Transactions on Parallel and Distributed Systems*, March 2000.

[6] CACI Products Company. *MODSIM III Reference Manual*, September 1997.

[7] R. Chamberlain, M. Franklin, R. Krchnavek, and B. Baysal. Design of an optically-interconnected multicomputer. In *Proc. of 5th Int'l Conf. on Massively Parallel Processing Using Optical Interconnections*, pages 114–122, June 1998.

[8] Roger Chamberlain, Ch'ng Shi Baw, Mark Franklin, Christopher Hackmann, Praveen Krishnamurthy, Abhijit Mahajan, and Michael Wrighton. Evaluating the performance of photonic interconnection networks. In *35th Annual Simulation Symposium*, April 2002.

[9] Roger Chamberlain, Mark Franklin, and Ch'ng Shi Baw. Gemini: An optical interconnection network for parallel processing. *IEEE Transactions on Parallel and Distributed Processing*, 13(10), October 2002.

[10] Roger Chamberlain, Mark Franklin, and Praveen Krishnamurthy. Optical network reconfiguration for signal processing applications. In *Proc. of the IEEE International*

Conference on Application-Specific Systems, Architectures and Processors, pages 344–355, 2002.

[11] Roger Chamberlain, Mark Franklin, and Abhijit Mahajan. VLSI photonic ring interconnect for embedded multicomputers: Architecture and performance. In *Proc. of 14th Conf. on Parallel and Distributed Computing Systems*, August 2001.

[12] M. Chateauneuf et al. Design, implementation and characterization of a 2-D bi-directional free-space optical link. In *Proc. of Optics in Computing*, pages 530–538, June 2000.

[13] Ch'ng Shi Baw. Design, analysis and simulation of optical interconnection networks. Master's thesis, Washington University, Saint Louis, MO, May 1999.

[14] Ch'ng Shi Baw, R.D. Chamberlain, and M.A. Franklin. Design of an interconnection network using VLSI photonics and free-space optical technologies. In *Proc. of 6th Int'l Conf. on Parallel Interconnects*, pages 52–61, October 1999.

[15] Ch'ng Shi Baw, R.D. Chamberlain, and M.A. Franklin. Fair scheduling in an optical interconnection network. In *Proc. of MASCOTS*, 1999.

[16] Paolo Cremonesi and Claudio Gennaro. Integrated performance models for SPMD applications and MIMD architectures. *IEEE Transactions on Parallel and Distributed Systems*, 13(7), 2002.

[17] Mark E. Crovella and Azer Bestavros. Self-similarity in World Wide Web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, 5(6), December 1997.

[18] J. Crow. Parallel optical interconnect - a cost performance breakthrough. In *LEOS proceedings*, pages 3–4. IEEE, 1996.

[19] L. Elada et al. Low-loss high-thermal-stability polymer interconnections for low-cost high performance massively parallel processing. In *Proc. of the Third International Conference on Massively Parallel Processing Using Optical Interconnections*, pages 192–205, October 1996.

[20] L. Fan and M.C. Wu. Optical interconnection network for massively parallel processors using beam-steering vertical cavity surface-emitting lasers. In *Proc. Second International Conference on Massively Parallel Processing Using Optical Interconnections(MPPOI)*, October 1995.

[21] Muralikrishna Gandluru. Optical networking and dense wavelength division multiplexing (DWDM), 1999.

[22] H. Scott Hinton. Progress in the smart pixel technologies. *IEEE Journal in Quantum Electronics*, 1996.

[23] TeraConnect Inc. Two Dimensional Opto-Electronics (DOE) - Shattering the Bandwidth Bottleneck, White Paper, 2002.

[24] J.L. Jewell, J.P. Harbison, A. Scherer, Y.H. Lee, and L.T. Florez. Vertical-cavity surface-emitting lasers: design, growth, fabrication, characterization. *IEEE Quantum Electron*, 27(2):1332, 1991.

[25] A. Kent and J. Williams. *Encyclopedia of Computer Science and Technology*. Cambridge University Press, 1999.

[26] F. Kiamilev, S. Esener, V. Ozgus, and S. Lee. Programmable optoelectronic multiprocessor systems. In *Digital Optical Computing*, volume CR35, pages 197–220. SPIE Press, July 1990.

[27] H. Kosaka et al. A two-dimensional optical parallel transmission using a vertical-cavity surface emitting laser array module and an image fiber. *IEEE Photon. Tech. Lett.*, 9:253–255, 1997.

[28] Abhijit Mahajan. Performance analysis of an optical interconnection network. Master's thesis, Washington University, Saint Louis, MO, 2000.

[29] Abhijit Mahajan, Mark Franklin, and Roger Chamberlain. Fairness issues in an embedded photonic ring interconnect. In *High Performance Embedded Computing Workshop*, september 2000.

[30] T. Maj et al. Interconnection of a two-dimensional array of vertical cavity surface emitting lasers to a receiver array via a fiber image guide. *Applied Optics*, 39:683–689, 2000.

[31] E. Markatos and T. LeBlanc. Shared memory multiprocessor trends and the implications for parallel program performance. Technical Report 420, University of Rochester, March 1992.

[32] M. Marsan et al. All-optical WDM multi-rings with differentiated QoS. *IEEE Communications Magazine*, pages 58–66, February 1999.

[33] Ethan L. Miller and Randy H. Katz. Input/output behavior of supercomputing applications. In *Proceedings of the conference on Supercomputing*, pages 567–576. ACM Press, 1991.

[34] Nils Nieuwejaar, David Kotz, Apratim Purakayastha, Carla Schlatter Ellis, and Michael Best. File-access characteristics of parallel scientific workloads. *IEEE Transactions on Parallel and Distributed Systems*, 7(10):1075–1089, 1996.

[35] J.A. O'Sullivan, M.A. Franklin, M.D. DeVore, and R.D. Chamberlain. Analysis of computational system performance in automatic target recognition. In *Proc. of High Performance Embedded Computing Workshop*, September 2000.

[36] Rajesh Kumar Pankaj. *Architectures for linear lightwave networks.* PhD thesis, Massachusetts Institute of Technology, 1992.

[37] Kihong Park, Gitae Kim, and Mark Crovella. On the relationship between file sizes, transport protocols, and self-similar network traffic. Technical Report 1996-016, Boston University, 30, 1996.

[38] V. Paxson and S. Floyd. Wide-area traffic: The failure of poisson modeling. *IEEE/ACM Transactions on Networking*, 3(3):226–244, 1994. http://ee.lbl.gov/nrg-papers.html.

[39] D. Plant et al. A 256 channel bi-directional optical interconnect using VCSELs and photodiodes on CMOS. In *Proc. of Optics in Computing*, pages 1046–1054, June 2000.

[40] D. V. Plant, B. Robertson, H. S. Hinton, M. H. Ayliffe, G. C. Boisset, W. Hsiao, D. Kabal, N. H. Kim, Y. S. Liu, M. R. Otazo, D. Pavlasek, A. Z. Shang, J. Simmons, K. Song, D. A. Thompson, and W. M. Robertson. A 4x4 VCSEL/MSM optical backplane demonstrator system. *Applied Optics*, 35:6365–6368, 1996.

[41] C. Qiao and R. Melham. Reconfiguration with time division multiplexed MINs for multiprocessor communications. *IEEE Transactions on Parallel and Distributed Systems*, 5(4), 1994.

[42] Chunming Qiao and Rami Melhem. Reconfiguration with time division multiplexed MIN's for multiprocessor communications. *IEEE Transactions on Parallel and Distributed Systems*, 5(4):337–352, 1994.

[43] Chunming M. Qiao, R. Melhem, D. Chiarulli, and S. Levitan. Dynamic reconfiguration of optically interconnected networks with time-division multiplexing. *Journal of Parallel and Distributed Computing*, 22(2):268–278, 1994.

[44] C. Salisbury and R. Melhem. Modeling compiled communication costs in multiplexed optical networks. In *Proceedings of IPPS*, pages 71–79.

[45] J. Sauer. A Multi-Gb/s opitcal interconnect. In *SPIE Proceeding, Digital Optical Computing II*, volume 1215, pages 198–207, 1990.

[46] M. Shreedhar and G. Varghese. Efficient fair queueing using deficit round robin. In *Proc. of SIGCOMM*, pages 231–243, August 1995.

[47] Walter Willinger, Murad S. Taqqu, Robert Sherman, and Daniel V. Wilson. Self-similarity through high-variability: statistical analysis of Ethernet LAN traffic at the source level. *IEEE/ACM Transactions on Networking*, 5(1):71–86, 1997.

[48] Carl W. Wilmsen, Henryk Temkin, and Larry A. Coldren. *Vertical-Cavity Surface-Emitting Lasers : Design, Fabrication, Characterization, and Applications*. Cambridge University Press, 1999.

Vita

Praveen Krishnamurthy

Date of Birth July 10, 1979

Place of Birth Raebareli, India

Degrees B.E. Electronics and Communication Engineering, May 2000
M.S. Computer Engineering, December 2002

Professional Societies Institute of Electrical and Electronics Engineers

Publications Roger Chamberlain, Ch'ng Shi Baw, Mark Franklin, Christopher Hackmann, Praveen Krishnamurthy, Abhijit Mahajan, and Micheal Wrighton, Evaluating the performance of photonic interconnection networks. In 35^{th} *Annual Simulation symposium*, April 2002.

Roger Chamberlain, Mark Franklin, and Praveen Krishnamurthy, Optical network reconfiguration for signal processing applications. In *Proc. of the IEEE International Conference on Application-Specific Systems, Architectures and Processors*, 2002.

December 2002